

دریافت مقاله: ۱۴۰۴/۰۶/۱۵

پذیرش مقاله: ۱۴۰۴/۱۰/۰۶

نوع مقاله: مروری

تفکر رایانشی به مثابه پلی میان علوم رایانه و ریاضیات

پویا کریمی

کارشناسی ارشد رشته آموزش ریاضی، بخش آموزش ریاضی، دانشکده ریاضی و کامپیوتر، دانشگاه شهید باهنر کرمان، کرمان، ایران
پست الکترونیکی: pouya.karimi1380@gmail.com

ابوالفضل رفیع پور*

دانشیار بخش آموزش ریاضی، دانشکده ریاضی و کامپیوتر، دانشگاه شهید باهنر کرمان، کرمان، ایران
پست الکترونیکی: drafiepour@gmail.com

چکیده

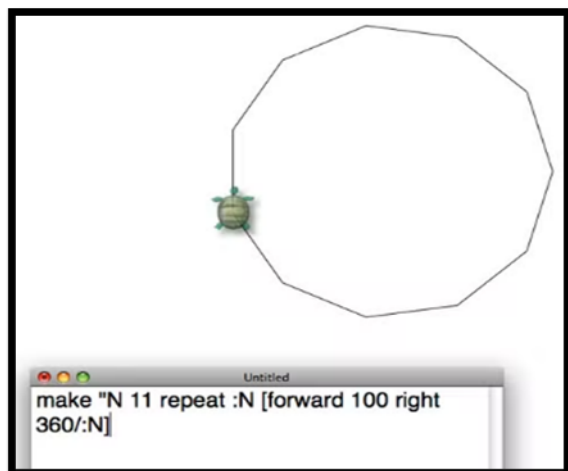
آن در آموزش ریاضی می‌پردازد. یافته‌ها نشان می‌دهد که ریاضیات، به‌واسطه تأکید بر الگوریتم، تجرید و بازنمایی، بستری طبیعی برای پرورش تفکر رایانشی فراهم می‌کند و ادغام اصول فنی علوم رایانه در آموزش ریاضی می‌تواند درک عمیق‌تر مفاهیم، توسعه مهارت‌های حل مسئله، و ارتقای خلاقیت و تفکر انتقادی دانش‌آموزان را به همراه داشته باشد. در نهایت، این مقاله بر ضرورت طراحی چارچوب‌های میان‌رشته‌ای تأکید می‌کند که بر پایه‌ی بنیان‌های اصیل علوم رایانه استوار بوده و در عین حال متناسب با نیازهای آموزشی به‌ویژه در سطح متوسطه طراحی شوند. بدین‌ترتیب، پرورش تفکر رایانشی در آموزش ریاضی نه‌تنها به بهبود یادگیری منجر خواهد شد، بلکه نقشی کلیدی در آماده‌سازی نسل آینده برای مواجهه با چالش‌های پیچیده عصر دیجیتال ایفا خواهد کرد.

کلیدواژه‌ها: تفکر رایانشی، علوم رایانه، ریاضیات، الگوریتم، یادگیری میان‌رشته‌ای

تفکر رایانشی در دو دهه اخیر به‌عنوان یکی از مفاهیم بنیادین علوم رایانه، جایگاهی محوری در توسعه فناوری‌های نوین یافته است. این مفهوم که در ایده‌های سیمور پاپرت ریشه دارد و توسط جنت وینگ به‌صورت نظام‌مند مطرح شد، فراتر از مهارت برنامه‌نویسی، رویکردی شناختی برای تحلیل و حل مسائل پیچیده در حوزه‌های مختلف علمی و اجتماعی تلقی می‌شود. با وجود گسترش پژوهش‌ها، مرور ادبیات نشان می‌دهد که شکاف قابل‌توجهی میان بنیان‌های نظری علوم رایانه مانند طراحی و تحلیل الگوریتم‌ها، نظریه محاسبه، کلان‌داده، محاسبات موازی و هوش مصنوعی و کاربردهای آموزشی آن در ریاضیات وجود دارد. پژوهش‌های فنی عمدتاً بر توسعه چارچوب‌های محاسباتی متمرکز بوده‌اند، در حالی که مطالعات آموزشی بیشتر بر ابزارهایی نظیر برنامه‌نویسی اسکرچ، پایتون و لوگو یا تلفیق سطحی تفکر رایانشی در برنامه‌های درسی تمرکز داشته‌اند. مطالعه حاضر یک مقاله مروری است که با هدف پر کردن این شکاف، به بررسی جایگاه تفکر رایانشی در علوم رایانه و تبیین ظرفیت‌های

مقدمه

تعداد زیادی از این لاک‌پشت‌ها — که گاهی «اسپرایت» نامیده می‌شوند — می‌توان انیمیشن‌ها و بازی‌های متنوع و خلاقانه‌ای ساخت.



شکل ۱. محیط برنامه نویسی لوگو

به تدریج، تفکر رایانشی به عنوان مجموعه‌ای از مؤلفه‌های بنیادین علوم رایانه تعریف شد؛ شامل تجزیه^۶، تشخیص الگو^۷، انتزاع^۸، طراحی الگوریتم^۹، و ارزیابی و اشکال‌زدایی^{۱۱} [۵]. این مؤلفه‌ها نه تنها زیربنای طراحی و تحلیل الگوریتم‌ها در علوم رایانشی هستند، بلکه امروزه در حوزه‌هایی چون هوش مصنوعی، علم داده، پردازش سیگنال و مدل‌سازی محاسباتی نیز کاربرد دارند. بنابراین، پرورش این نوع تفکر در محیط‌های آموزشی به طور مستقیم با نیازهای اصلی علوم رایانشی همسو است.

دنینگ [۸] و دنینگ و تدری^{۱۲} [۹] کوشیده‌اند تا مرزهای دقیق تفکر رایانشی را در ارتباط با نظریه محاسبه، الگوریتم‌ها و مدل‌های محاسباتی ترسیم کنند. کلارک، گرومبرگ و پلید^{۱۳} [۱۰] با ارایه ابزارهای بررسی مدل نشان دادند که چگونه مؤلفه‌های تفکر رایانشی در روش‌های صوری و خودکارسازی صحت‌یابی سامانه‌ها

در دو دهه اخیر، یکی از مفاهیم محوری که به سرعت جایگاهی بنیادین در علوم رایانه، آموزش و سیاست‌گذاری‌های جهانی یافته است، تفکر رایانشی است. این اصطلاح که نخستین بار توسط جنت وینگ [۲، ۱] به صورت نظام‌مند مطرح شد، امروز نه صرفاً به عنوان مهارت برنامه‌نویسی، بلکه به مثابه یک الگوی شناختی و محاسباتی برای تحلیل و حل مسائل پیچیده در حوزه‌های گوناگون علمی و اجتماعی شناخته می‌شود. در واقع، تفکر رایانشی پلی میان نظریه محاسبات، طراحی الگوریتم‌ها، و کاربردهای میان‌رشته‌ای آن در آموزش، مهندسی و علوم داده ایجاد می‌کند [۳].

پیشینه نظری این رویکرد به تلاش‌های سیمور پاپرت^۲ در توسعه زبان برنامه‌نویسی لوگو^۳ در دهه ۱۹۶۰ باز می‌گردد. لوگو نمونه‌ای از یک محیط آموزشی بود که کودکان می‌توانستند با آن از طریق دستورات ساده، مفاهیم الگوریتمی و منطقی را تجربه کنند [۴]. این محیط‌های آموزشی، نخستین نمونه‌های کاربردی پیوند علوم رایانه با آموزش ریاضی محسوب می‌شوند و به نوعی مبنای شکل‌گیری ابزارهای نوین‌تر مانند برنامه نویسی اسکرچ^۴ و جئو جبرا^۵ شدند [۵]. فعالیت‌های مرتبط با برنامه‌نویسی لوگو در حوزه‌هایی چون ریاضیات، زبان، موسیقی، رباتیک، مخابرات و علوم کاربرد دارند. محیط‌های محبوب برنامه‌نویسی لوگو، شامل «لاک‌پشت» بوده‌اند؛ که در ابتدا یک موجود رباتیک بر روی زمین بود و با تایپ دستورات در کامپیوتر، قادر به حرکت بود. به سرعت، لاک‌پشت به صفحه گرافیک کامپیوتر منتقل شد، جایی که می‌توانست اشکال، طرح‌ها و تصاویر مختلف را رسم کند [۵]. برخی گونه‌های لاک‌پشت‌ها توانایی تغییر شکل به شکل‌های مختلفی مانند پرنده، ماشین، هواپیما یا هر چیزی که طراح تصور می‌کند، را دارند. در محیط‌های لوگو، با وجود

6-Sprit
7-Decomposition
8-Pattern Recognition
9-Abstraction
10-Algorithm Design
11-Debugging & Evaluation
12-Denning & Tedre
13-Clarke, Grumberg, & Peled

1-Wing
2-Seymour papert
3-Logo
4-Scratch
5-Geogebra

با وجود گسترش روزافزون پژوهش‌ها در زمینه تفکر رایانشی، مرور انتقادی ادبیات نشان می‌دهد که شکاف‌های معناداری میان دو حوزه علوم رایانه و آموزش همچنان باقی است. در ادبیات علوم رایانه، تفکر رایانشی عمدتاً در قالب مفاهیم نظری که گفته شد، مورد بررسی قرار گرفته است. این مطالعات [۱۱، ۱۲، ۱۳] نشان داده‌اند که تفکر رایانشی بنیانی برای توسعه چارچوب‌های نظری و ابزارهای محاسباتی محسوب می‌شود اما کمتر به چگونگی انتقال این مفاهیم به عرصه آموزش و یادگیری در مقاطع مختلف تحصیلی پرداخته‌اند.

از منظر علوم رایانه، ریاضیات بستری طبیعی برای ادغام تفکر رایانشی فراهم می‌کند. ریاضی ذاتاً بر الگوریتم، انتزاع و بازنمایی استوار است و بسیاری از مسائل ریاضی را می‌توان به صورت محاسباتی مدل‌سازی و حل کرد. این هم‌افزایی به‌ویژه در چارچوب آموزش استم^{۱۴} (علوم، فناوری، مهندسی و ریاضیات) اهمیت دارد، جایی که تفکر رایانشی به عنوان عامل پیونددهنده میان علوم محاسباتی و ریاضی عمل می‌کند [۱۵]. به طور کلی، پیشینه پژوهش بیانگر آن است که تفکر رایانشی هم به عنوان شالوده‌ای مفهومی در علوم رایانه و هم به عنوان ابزاری میان‌رشته‌ای در آموزش و یادگیری مطرح شده است. با وجود این، شکاف معناداری میان دو حوزه وجود دارد: بخش عمده‌ای از پژوهش‌های علوم رایانه بر توسعه چارچوب‌های نظری و فنی تمرکز دارند، در حالی که پژوهش‌های آموزشی بیشتر به کاربردهای کلاسی و یادگیری دانش‌آموزان پرداخته‌اند. این مقاله می‌کوشد پلی میان این دو حوزه برقرار کند و نشان دهد که چگونه می‌توان با بهره‌گیری از بنیان‌های علوم رایانه‌ای، تفکر رایانشی را در آموزش ریاضی به شیوه‌ای نظام‌مند و اصولی به کار گرفت. بنابراین، ادغام هدفمند تفکر رایانشی در آموزش ریاضی می‌تواند هم‌زمان به تقویت توانمندی‌های حل مسئله، خلاقیت و تفکر انتقادی دانش‌آموزان و نیز به ارتقای شایستگی‌های محاسباتی

متجلی می‌شوند. در همین راستا، پژوهش‌های دیگری در زمینه طراحی الگوریتم‌ها [۱۱]، معماری‌های موازی و چارچوب‌های پردازش کلان‌داده [۱۲]، و نیز مدل‌های داده‌ای نوین [۱۳] به تقویت این بنیان کمک کرده‌اند. این مطالعات نشان می‌دهند که تفکر رایانشی صرفاً یک مهارت عمومی نیست، بلکه در قلب علوم رایانه و در لایه‌های مختلف از نظریه تا کاربرد جایگاه دارد.

سیر تکامل نظری این مفهوم نشان می‌دهد که تفکر رایانشی از کاربست ابزارهای فناورانه فراتر رفته و به نوعی ذهنیت تحلیلی و ساختاریافته برای مواجهه با پیچیدگی‌ها تبدیل شده است. این ذهنیت متکی است بر مؤلفه‌هایی چون تجزیه مسئله به اجزای ساده‌تر، تشخیص الگوها، انتزاع‌سازی، طراحی الگوریتم‌های کارآمد، ارزیابی و بهینه‌سازی راه‌حل‌ها، و تعمیم‌پذیری در موقعیت‌های نوین است [۱۴]. این مؤلفه‌ها که خاستگاه اصلی آنها در علوم رایانه است، اکنون در حوزه‌های مختلف، از جمله آموزش ریاضی، جایگاهی محوری یافته‌اند. در جهانی که فناوری نقش محوری در آن ایفا می‌کند و داده‌ها بخش عظیمی از تصمیم‌گیری‌های فردی و اجتماعی را شکل می‌دهند، مهارتی مانند تفکر رایانشی به عنوان یکی از شایستگی‌های ضروری قرن بیست و یکم در نظر گرفته می‌شود [۲]. امروزه بسیاری از نظام‌های آموزشی در سراسر جهان، با آگاهی از این ضرورت، دست به بازطراحی برنامه‌های درسی زده‌اند تا با پرورش مهارت‌های تحلیلی، انتقادی و مسئله‌محور، دانش‌آموزان را برای مشارکت فعال در جامعه دیجیتال آماده کنند. نمونه‌هایی از این تلاش‌ها را می‌توان در پیاده‌سازی رسمی تفکر رایانشی در برنامه‌های درسی کشورهایی مانند انگلستان (۲۰۱۴)، آلمان، کره جنوبی، چین و تایوان (از ۲۰۱۶) مشاهده کرد. این حرکت جهانی حاکی از آن است که تفکر رایانشی دیگر یک مهارت تخصصی محدود به علوم رایانه نیست، بلکه به مثابه یک مهارت جهانی قابل انتقال و عامل کلیدی در رقابت‌پذیری اقتصادی، نوآوری فناورانه و پیشرفت اجتماعی تلقی می‌شود.

۱۴- در این آموزش یکپارچه چهار رشته علوم، فناوری، ریاضیات و مهندسی با یکدیگر ادغام شده و فرایند یاددهی و یادگیری شکل می‌گیرد.

مورد نیاز علوم رایانشی بینجامد. مقاله حاضر با هدف بررسی این پیوند میان علوم رایانه و آموزش ریاضی نگاشته شده است. در این پژوهش، ضمن مرور پیشینه نظری تفکر رایانشی و معرفی چارچوب‌های محاسباتی آن، ظرفیت‌های آموزشی این مهارت در کلاس‌های ریاضی و علوم رایانه تحلیل می‌شود و راهکارهایی برای طراحی فعالیت‌های میان‌رشته‌ای و توسعه ابزارهای نرم‌افزاری آموزشی ارائه خواهد شد.

۲- تعریف تفکر رایانشی

اولین تلاش‌ها برای تعریف تفکر رایانشی در اوایل دهه ۱۹۰۰ برجسته شد که در چارچوب فرایندهای تفکر الگوریتمی و صورت‌بندی شناختی مسئله بیان شدند [۳]. در سال ۱۹۹۶، اصطلاح تفکر رایانشی توسط پاپرت پیشنهاد شد و پس از آن مطالعات جنت وینگ (۲۰۰۶) انجام شد. سپس، به تدریج به موضوعی مورد توجه در محافل علمی تبدیل شد [۱۶]. اگرچه در متون علمی اجماع جهانی در مورد چارچوب‌های تفکر رایانشی وجود ندارد، اما تلاش‌های متعددی برای تعریف آن صورت گرفته است [۱۸، ۱۷]. تفکر رایانشی به عنوان یکی از مؤلفه‌های بنیادین آموزش تلفیقی استم، به‌ویژه در تعامل بین علوم کامپیوتر و ریاضیات، جایگاه ویژه‌ای یافته است. بیش از پنج دهه پیش، سیمور پاپرت رویکردی نوین برای آموزش ریاضی ارائه داد که بر اساس برنامه‌نویسی رایانه‌ای بود. او بر اهمیت فعالیت‌های ریاضی تأکید کرد که باید شخصی، خلاق و کنجکاوانه باشند. این نوع فعالیت‌ها ترکیبی از یک زبان رسمی (زبان برنامه نویسی لوگو)، دقت در ریاضیات و دریافت بازخورد کامپیوتری هستند. نکته جالب توجه این است که سیمور پاپرت به‌عنوان یک ریاضیدان که تحت تأثیر نظریه‌های ژان پیاژه قرار داشت، به تحقیق در زمینه یادگیری کودکان خردسال و چگونگی تقویت این فرایندها توسط رایانه‌ها علاقه‌مند بود [۱۹].

ایده‌های پاپرت، اگرچه در زمان خود نوآورانه

بودند، اما مفهوم «تفکر رایانشی» به شکل امروزی آن، با کارهای جنت وینگ در سال ۲۰۰۶ به طور گسترده‌تری مطرح شد. وینگ (۲۰۰۶) استدلال کرد که تفکر رایانشی یک مهارت اساسی است که همه باید آن را بیاموزند، نه فقط متخصصان رایانه. او تفکر رایانشی را به عنوان «فرایند فکری درگیر در فرموله کردن یک مسئله و بیان راه‌حل‌های آن به گونه‌ای که بتواند به طور موثر توسط یک عامل پردازش اطلاعات (انسان یا ماشین) انجام شود» تعریف کرد. در سال‌های پس از آن، مفهوم تفکر رایانشی به تکامل خود ادامه داد. وینگ (۲۰۰۸) بر اهمیت آن در تمامی رشته‌ها و زندگی روزمره تأکید کرد. سازمان‌هایی نظیر انجمن معلمان علوم کامپیوتر^{۱۵} و انجمن ماشین‌های رایانشی^{۱۶} (۲۰۱۷) نیز تعاریف جامع‌تری از تفکر رایانشی ارائه دادند که شامل مولفه‌هایی چون تجزیه، تشخیص الگو، انتزاع، طراحی الگوریتم، ارزیابی و تعمیم می‌شود. در سال‌های اخیر، پژوهش‌ها بر ادغام تفکر رایانشی در برنامه‌های درسی مختلف و توسعه ابزارهای آموزشی برای تقویت آن متمرکز شده است. برای مثال، محیط‌های برنامه‌نویسی بصری مانند اسکرچ به‌طور گسترده‌ای برای آموزش مفاهیم تفکر رایانشی به دانش‌آموزان در سنین مختلف مورد استفاده قرار گرفته‌اند [۲۰]. همچنین، تلاش‌هایی برای ارزیابی و سنجش میزان تفکر رایانشی دانش‌آموزان صورت گرفته است [۲۱]. تا سال ۲۰۲۵، تفکر رایانشی نه تنها به‌عنوان یک مهارت کلیدی در علوم کامپیوتر، بلکه به‌عنوان یک شایستگی ضروری برای مواجهه با چالش‌های دنیای پیچیده و مبتنی بر داده‌ها، بیش از پیش مورد تأکید قرار گرفته است. پژوهش‌ها همچنان بر چگونگی ادغام موثر آن در آموزش‌های مختلف و در سطوح گوناگون تحصیلی ادامه دارد [۲۲].

۳- مؤلفه‌های تفکر رایانشی

تفکر رایانشی را می‌توان مجموعه‌ای از مهارت‌ها و

تفکر رایانشی را با هدف تقویت آموزش و یادگیری معنادار ریاضیات در محیط‌های آموزشی مختلف، پذیرا باشند.

۳-۱ تجزیه

نخستین گام در حل مسائل رایانشی، تجزیه یک مسئله پیچیده به زیربخش‌های ساده‌تر است. این مؤلفه با مفاهیم «ماژولار بودن» در طراحی نرم‌افزار و «ساختمان داده‌ها» در علوم رایانه پیوند دارد [۹].

تجزیه، به یادگیرندگان کمک می‌کند تا یک مسئله پیچیده را به بخش‌های کوچک‌تر تقسیم کنند، تا حل آنها ساده‌تر گردد. با شناسایی بخش‌های کلیدی مسئله، تعیین روابط بین آنها، و تدوین برنامه‌ای برای مقابله با هر بخش، افراد می‌توانند از ترس کمتری نسبت به مسئله برخوردار شوند و اعتماد به نفس بیشتری در توانایی خود بیابند [۲۵]. یادگیری نحوه استفاده از تجزیه، به یادگیرندگان کمک می‌کند مسائل پیچیده را قابل مدیریت‌تر سازند و راهی برای همکاری و کار گروهی در بخش‌های مختلف یک مسئله فراهم می‌کند، قبل از آن که دنبال کردن قطعات و یافتن راه حل کلی صورت گیرد. برای مثال، در توسعه یک نرم‌افزار پردازش تصویر، مسئله اصلی به زیربخش‌هایی مانند خواندن داده، پیش‌پردازش، فیلترگذاری و استخراج ویژگی تجزیه می‌شود. این فرایند همان چیزی است که در آموزش ریاضی می‌تواند در تجزیه مسائل بزرگ به گام‌های کوچک‌تر برای حل الگوریتمیک بازتاب یابد.

۳-۲ تشخیص الگو

شناسایی شباهت‌ها و تفاوت‌ها در داده‌ها، الگوریتم‌ها و ساختارها، هسته بسیاری از شاخه‌های علوم رایانه مانند یادگیری ماشین^{۱۷} و بینایی ماشین^{۱۸} است [۲۷، ۲۶، ۲۷]. در سطح نظری، تشخیص الگو مبنای توسعه الگوریتم‌های کارآمد برای طبقه‌بندی و خوشه‌بندی داده‌هاست. از منظر آموزشی، تشخیص الگو در ریاضیات (مثلاً دنباله‌های

فرایندهای شناختی دانست که ریشه در علوم رایانه دارند و به انسان‌ها امکان می‌دهند مسائل پیچیده را به صورت الگوریتمی تحلیل، طراحی و حل کنند [۳، ۱]. برخلاف رویکردهای عمومی که تفکر رایانشی را صرفاً ابزار آموزشی معرفی می‌کنند، در حوزه علوم رایانه این مفهوم ارتباطی مستقیم با اصولی همچون نظریه محاسبه، طراحی الگوریتم‌ها، پیچیدگی محاسباتی، و معماری نرم‌افزار دارد. در ادامه، مهمترین مؤلفه‌های این نوع تفکر را از منظر علوم رایانه و ریاضیات تشریح می‌شوند. توانایی بهره‌گیری از راهبردهای تفکر رایانشی برای موفقیت در عصر حاضر، که بسیاری از مشاغل نیازمند مهارت حل مسئله، تجزیه و تحلیل دقیق و تفسیر داده‌ها است، اهمیت فراوانی دارد [۲۳]. با تکیه بر راهبردهای تفکر رایانشی، به یادگیرندگان کمک می‌شود تا رویکردی منظم و مؤثر در حل مسائل ایجاد کنند، رویکردی که در داخل و خارج از کلاس درس قابل کاربرد است و آنان را در موفقیت‌های تحصیلی و آمادگی برای مشاغل آینده یاری می‌دهد. با ارتباط دادن مسائل ریاضی به موقعیت‌های واقعی زندگی و امتحان کردن روش‌ها و راهبردهای مختلف حل مسئله، دانش‌آموزان درک عمیق‌تری از مفاهیم پیدا می‌کنند و خود را به عنوان یادگیرندگان مادام‌العمر می‌نگرند [۲]. با تکیه بر چارچوب‌های مطرح‌شده در زمینه تفکر رایانشی، محققان تلاش کرده‌اند تا این مفهوم را در حوزه‌های موضوعی مختلف، به ویژه آموزش ریاضیات و علوم رایانه، تعریف و بررسی کنند. اگرچه تعاریف متعددی برای تفکر رایانشی ارائه شده است، مفاهیمی مانند تشخیص الگو، انتزاع، تجزیه، و تفکر الگوریتمی در تمامی این تعاریف مشترک هستند و ما علاوه بر این مؤلفه‌ها اشکال‌زدایی و تعمیم‌پذیری را نیز شرح می‌دهیم. همچنین تفکر رایانشی را می‌توان در ریاضیات، علوم رایانه، مهندسی و زندگی روزمره به کار برد و از آن می‌توان برای یافتن راه حل برای مشکلات پیچیده در جامعه استفاده کرد [۲۴]. نیاز مبرمی وجود دارد که فراگیران و معلمان به طور کامل ظرفیت‌های آموزشی

نیز همین رویکرد در ساده‌سازی مسائل و استفاده از نمادگذاری‌ها دیده می‌شود.

این مهارت به یادگیرندگان کمک می‌کند مهمترین اطلاعات موجود در یک مسئله را شناسایی کنند و در عین حال، جزئیات نامربوط را نادیده بگیرند همچنین این امکان را می‌دهد تا در فرایند حل مسئله، تفکر انتقادی داشته باشند و تمرکز خود را بر روی موارد ضروری قرار دهند [۲۵]. این مهارت به توانایی استدلال منطقی برای تحلیل مسائل و یافتن راه‌حل‌های قابل اجرا اشاره دارد. افراد با استفاده از این مهارت می‌توانند اطلاعات را سازمان‌دهی کرده و الگوهای منطقی را برای پیش‌بینی نتایج به کار گیرند. به عنوان مثال، در برنامه‌نویسی اسکرچ، می‌توان از شرط‌های منطقی برای تعیین نحوه اجرای یک انیمیشن بهره برد. یادگیری نحوه به‌کارگیری انتزاع، به دانش‌آموزان کمک می‌کند تا راهبردهای موثری برای حل مسائل توسعه دهند. وقتی دانش‌آموزان بازنمایی شخصی از مسائل ارائه می‌دهند، آنها مسائل را از دید خودشان درک می‌کنند. این کار موجب معنادارتر شدن مطالب برای آنان می‌شود و کمک می‌کند تا راهبردهایی را ایجاد و توسعه دهند که برایشان از نظر ذهنی و عملی، معنا دار باشد [۱۷].

۳-۴ طراحی الگوریتم

شاید بنیادی‌ترین مؤلفه تفکر رایانشی طراحی الگوریتم باشد. الگوریتم‌ها نه تنها هسته اصلی علوم رایانه محسوب می‌شوند، بلکه در موضوعاتی مانند بهینه‌سازی، هوش مصنوعی، محاسبات ابری، و الگوریتم‌های موازی نیز کاربرد مستقیم دارند [۱۱]. توانایی طراحی الگوریتم‌های گام‌به‌گام، ارزیابی پیچیدگی زمانی و مکانی آنها، و انتخاب بهترین روش حل، همان چیزی است که تفکر رایانشی را از مهارت‌های عمومی به یک بنیان محاسباتی ارتقا می‌دهد الگوریتم‌ها رویکردهای گام به گام منطقی و هدفمندی هستند که دانش‌آموزان می‌توانند برای مقابله با انواع مسائل از آنها استفاده کنند. این مؤلفه که شامل توسعه

عددی) می‌تواند مقدمه‌ای برای درک مفاهیم پیشرفته‌تر مانند مدل‌سازی داده یا تحلیل پیچیدگی باشد.

این مؤلفه تفکر رایانشی شامل شناسایی شباهت‌ها و ساختارهای مشترک در مسائل مختلف است. یادگیرندگان با استفاده از این مهارت می‌توانند الگوریتم‌های عمومی ایجاد کرده و آنها را برای حل مسائل مشابه به کار گیرند. برای مثال، طراحی حرکت نویسه‌ها در یک بازی اسکرچ می‌تواند از الگوهای حرکتی استاندارد پیروی کند. یادگیرندگان از تشخیص الگوها برای یافتن راه‌هایی که ممکن است مسائل مشابه یا متفاوتی باشند، استفاده می‌کنند. این مهارت به آنها کمک می‌کند تا به سرعت تشخیص دهند چگونه مسائل جدید، با توجه به مسائل پیشین، قابل حل هستند. تشخیص الگوها ارتباط میان مسائل را برقرار کرده، و از این‌رو، پشتیبانی مهمی در ارائه راه‌حلی بر اساس دانش قبلی محسوب می‌شود [۲۵]. یادگیری تشخیص الگوها برای درک عمیق‌تر مفاهیم و ایجاد ارتباط میان ایده‌های مختلف بسیار حائز اهمیت است. همچنین، این مهارت به دانش‌آموزان کمک می‌کند تا ببینند چطور دانش ریاضی آنها می‌تواند در زندگی خارج از مدرسه نیز مفید و کاربردی باشد. هدف تشخیص الگو، ساده‌سازی بیشتر مسئله با یافتن نقاط مشترک یا تفاوت‌ها است که به درک عمیق‌تر مسئله پیچیده به طور کلی کمک می‌کند. این می‌تواند منجر به راه‌حل‌های کارآمدتر و تعمیم‌پذیرتری شود.

۳-۳ انتزاع

انتزاع به معنای نادیده گرفتن جزئیات غیرضروری و تمرکز بر ویژگی‌های کلیدی یک سیستم یا مسئله است. این مؤلفه اساس طراحی زبان‌های برنامه‌نویسی سطح بالا، و حتی نظریه اتوماتا^{۱۹} را شکل می‌دهد. در علوم رایانه، انتزاع باعث می‌شود بتوان سیستم‌های بسیار پیچیده مانند سیستم‌عامل‌ها یا شبکه‌های رایانه‌ای را در قالب لایه‌ها و مدل‌های مفهومی مدیریت کرد. در ریاضیات آموزشی

باشد، اما بازخورد محترمانه و سازنده از طرف اساتید، معلمان و همسالان می‌تواند بخش‌های اساسی و اثرگذار در فرایند یادگیری باشد. پشتکار در این مسیر کمک می‌کند تا دانش‌آموزان اعتماد بیشتری به توانایی‌های خود پیدا کنند و از تکرار اشتباهاتشان جلوگیری نمایند.

فرایند اشکال‌زدایی در برنامه‌نویسی، یک تمرین عالی برای تفکر انتقادی و حل مسئله است. وقتی برنامه‌ای درست کار نمی‌کند، دانش‌آموز باید گام به گام کدهای خود را بررسی کرده، خطاها را شناسایی کند و راه‌حل‌هایی برای اصلاح آنها بیابد. این مهارت‌ها مستقیماً به ریاضیات قابل تعمیم هستند. در ریاضیات نیز، وقتی دانش‌آموزی در حل یک مسئله به بن‌بست می‌رسد یا به پاسخ اشتباه می‌رسد، باید مراحل حل خود را مرور کند، خطای منطقی یا رایانشی را پیدا کند و راه‌حل را اصلاح کند. این شباهت‌ها، تفکر رایانشی را به یک ابزار قدرتمند برای تقویت مهارت‌های اشکال‌زدایی در ریاضیات تبدیل می‌کند [۲۱].

۳-۶ تعمیم‌پذیری

تعمیم‌پذیری در تفکر رایانشی به معنای توانایی به‌کارگیری یک راه‌حل، الگوریتم یا الگوی محاسباتی در موقعیت‌ها و مسائل جدید است. این مهارت در علوم رایانه اهمیت بنیادی دارد، زیرا بسیاری از پیشرفت‌های این حوزه نه از طریق طراحی الگوریتم‌های کاملاً جدید، بلکه با بازتولید و انطباق الگوریتم‌های موجود در زمینه‌های مختلف به‌دست آمده‌اند [۳۰]. قابلیت تعمیم یک راه‌حل به مسائل جدید یا دامنه‌های متفاوت، نشان‌دهنده بلوغ تفکر رایانشی است. این مؤلفه با مفاهیم انتقال یادگیری در هوش مصنوعی و باز به‌کارگیری کد در مهندسی نرم‌افزار ارتباط دارد [۳۰]. در حوزه آموزش ریاضی، تعمیم‌پذیری می‌تواند به درک عمیق‌تر روابط ریاضی و کاربرد آنها در موقعیت‌های نوین کمک کند.

از دیدگاه نظریه محاسبات، تعمیم‌پذیری با مفهوم

یک راه‌حل واضح و گام به گام یا مجموعه‌ای از قوانین برای یک مسئله است که می‌تواند تکرار شود تا به یک نتیجه قابل پیش‌بینی و قابل اعتماد دست یابد [۲۴]. در زمینه‌های رایانشی مدرن، این راه‌حل معمولاً به شکل یک دنباله دقیق از دستورالعمل‌های منطقی و مرتب است که برای اجرا توسط رایانه طراحی شده است، اگرچه می‌تواند توسط انسان نیز انجام شود [۲۹]. تفکر الگوریتمی شامل توانایی طراحی و توسعه مجموعه‌ای از دستورالعمل‌های دقیق برای حل مسائل است. این مهارت در برنامه‌نویسی و ریاضیات به دانش‌آموزان کمک می‌کند تا مسائل را به روش‌های مرحله به مرحله حل کنند (فوک ۲۰۰۶). در محیطی مانند اسکرچ، تفکر الگوریتمی هنگام طراحی دنباله‌ای از دستورات برای ایجاد یک بازی تعاملی مشاهده می‌شود.

۳-۵ اشکال‌زدایی و ارزیابی

یکی از مؤلفه‌های کلیدی تفکر رایانشی، توانایی شناسایی خطاها و اصلاح آنها در فرایند حل مسئله است. در علوم رایانه، اشکال‌زدایی نه تنها در سطح برنامه‌نویسی بلکه در اعتبارسنجی الگوریتم‌ها، آزمون نرم‌افزار و تحلیل امنیت داده‌ها نیز مطرح است [۱۸]. این مؤلفه با فرایند «اعتبارسنجی ریاضی» شباهت دارد و باعث توسعه مهارت‌های خودبازبینی و بهینه‌سازی در دانش‌آموزان می‌شود.

اشکال‌زدایی به یادگیرندگان کمک می‌کند تا اشتباهات را پیدا کرده و آنها را اصلاح کنند، تا بتوانند کار خود را بهبود بخشند. این فرایند شامل آزمایش راه‌حل‌های مختلف بر روی بخش‌های کوچک‌تر مسئله است، تا مشخص شود در کجا ممکن است خطا صورت گرفته باشد. زمانی که افراد یاد می‌گیرند چگونه کارهای خود را مرور، اصلاح و بهبود بخشند، در حل مسائل، آموزش‌های عمومی و سایر حوزه‌ها به مهارت‌های بهتری دست می‌یابند [۲۵]. هر چند اشکال‌زدایی ممکن است فرایندی خسته‌کننده

بعدها در طراحی الگوریتم‌های تحلیل سری‌های زمانی یا پردازش داده‌های حسگر به کار رود. علاوه بر این، چارچوب‌های دیگری نیز برای درک تفکر رایانشی ارائه شده است. برنان و رسنیک (۲۰۱۲) آن را در ابعاد مفاهیم، شیوه‌ها و دیدگاه‌های رایانشی بررسی می‌کنند. وینتروپ^{۲۷} و همکاران (۲۰۱۶) چهار دسته اصلی شامل روش‌های کار با داده‌ها، مدل‌سازی و شبیه‌سازی، روش‌های حل مسئله رایانشی و تفکر سیستمی را مطرح می‌کنند. این چارچوب همانطور که در جدول ۱ مشاهده می‌کنید، یکی از مدل‌های کلیدی برای تحلیل و به‌کارگیری تفکر رایانشی در آموزش ریاضیات و علوم است. این چارچوب به معلمان و دانش‌آموزان کمک می‌کند تا تفکر رایانشی را به‌صورت کاربردی در فرایند یادگیری ادغام کنند.

جدول ۱. مدل مفهومی تفکر رایانشی [۷]	
دسته‌بندی	توضیح
روش‌های کار با داده‌ها	جمع‌آوری، تحلیل، پردازش و نمایش داده‌ها برای استخراج اطلاعات معنادار (مانند نمودارها و ابزارهای تجزیه و تحلیل).
مدل‌سازی و شبیه‌سازی	ایجاد مدل‌های ریاضی یا کامپیوتری برای شبیه‌سازی و تحلیل پدیده‌های پیچیده (مانند پیش‌بینی آب‌وهوا یا سیستم‌های زیستی).
روش‌های حل مسئله رایانشی	طراحی و پیاده‌سازی راه‌حل‌های الگوریتمی برای مسائل علمی و ریاضی (مانند برنامه‌نویسی برای حل معادلات یا بهینه‌سازی).
تفکر سیستمی	بررسی تعاملات و ارتباطات بین اجزای یک سیستم برای درک رفتار کلی آن (مانند مطالعه اکوسیستم‌ها).

همچنین لی و همکاران (۲۰۲۰) بر مفاهیم دنباله‌ها، تکرار و شرط‌ها در ساختار برنامه‌نویسی تاکید دارند، در حالی که هویلس و ناس^{۲۸} (۲۰۱۵) انتزاع، تفکر الگوریتمی و شناسایی الگوها را مورد توجه قرار می‌دهند. بوکانی^{۲۹} و همکاران (۲۰۱۶) نیز مؤلفه‌هایی چون انتزاع، تفکر الگوریتمی، تجزیه، اشکال‌زدایی و تعمیم را برجسته می‌کنند. در جدیدترین تلاش برای ایجاد یک ساختار منسجم از مؤلفه‌های تفکر رایانشی در آموزش ریاضیات

«قابلیت باز به‌کارگیری^{۲۱}» و «قابلیت انتقال^{۲۲}» گره خورده است. یک الگوریتم زمانی ارزش واقعی پیدا می‌کند که بتوان آن را در دامنه‌های گوناگون به کار گرفت. به‌عنوان نمونه، الگوریتم تقسیم و غلبه که ابتدا برای مرتب‌سازی مطرح شد، بعدها در مسائل هندسی محاسباتی، پردازش سیگنال و حتی شبیه‌سازی‌های علمی به‌کار گرفته شد [۱۱]. در سطح مهندسی نرم‌افزار، تعمیم‌پذیری اساس طراحی ماژولار و شیء‌گرایی نرم‌افزار است. استفاده از الگوهای طراحی^{۲۳} نمونه‌ای از تلاش برای ایجاد ساختارهایی است که در پروژه‌های متعدد و محیط‌های گوناگون قابل استفاده باشند. این موضوع نشان می‌دهد که تعمیم‌پذیری نه تنها مهارتی شناختی، بلکه یک اصل فنی در توسعه سیستم‌های نرم‌افزاری است. در حوزه یادگیری ماشین و هوش مصنوعی، تعمیم‌پذیری یکی از معیارهای اصلی کارآمدی الگوریتم‌ها محسوب می‌شود. الگوریتمی موفق است که نه فقط داده‌های آموزش^{۲۴} بلکه داده‌های جدید و ناشناخته را نیز با دقت مناسب تحلیل کند. این همان مفهوم «قابلیت تعمیم‌پذیری^{۲۵}» در یادگیری ماشین است [۳۱]. چالش «بیش‌برازش^{۲۶}» دقیقاً ناشی از ناتوانی مدل در تعمیم‌پذیری به داده‌های جدید است. در علوم داده و محاسبات علمی، تعمیم‌پذیری به پژوهشگران اجازه می‌دهد که مدل‌های ریاضی و محاسباتی خود را به دامنه‌های جدید گسترش دهند. برای مثال، روش‌های عددی توسعه‌یافته در حل معادلات دیفرانسیل معمولی بعدها برای معادلات دیفرانسیل با مشتقات جزئی و مدل‌های پیچیده‌تر مورد استفاده قرار گرفتند.

از منظر آموزشی، پرورش تعمیم‌پذیری در دانش‌آموزان به این معناست که آنان یاد بگیرند یک راه‌حل ریاضی یا الگوریتمی را به مسائل مشابه در علوم دیگر منتقل کنند. به‌عنوان نمونه، درک الگوی دنباله‌های عددی می‌تواند

21-Reusability
22-Transferability
23-Design Patterns
24-Training Data
25-Generalization Ability
26-Overfitting

27-Weintrop
28-Hoyles & Noss
29-Bocconi

منطق زمانی بازنمایی می‌شود و در نهایت، صحت سیستم به صورت خودکار بررسی چنین روندی نشان می‌دهد که مسیر انتزاع تا خودکارسازی صحت‌یابی، جلوه طبیعی تفکر رایانشی در علوم رایانه است [۳۳، ۱۰].

در حوزه ساختمان داده‌ها و الگوریتم‌ها نیز مؤلفه‌های اساسی تفکر رایانشی آشکار می‌شوند. تجزیه مسئله، تشخیص الگو، انتزاع‌سازی و طراحی الگوریتم دقیقاً همان فرایندهایی هستند که در انتخاب ساختمان داده مناسب و طراحی الگوریتم‌های کارآمد نمود پیدا می‌کنند. روش‌های کلاسیک طراحی الگوریتم‌ها، همچون تقسیم و غلبه، حریصانه و برنامه‌ریزی پویا، همگی صورت‌بندی‌های گوناگون این مؤلفه‌ها محسوب می‌شوند. افزون بر این، تحلیل مرتبه زمانی و مقایسه کارایی میان طرح‌های مختلف، جنبه ارزیابانه تفکر رایانشی را در علوم رایانه آشکار [۱۱]. در مهندسی نرم‌افزار، انتزاع‌های لایه‌ای، پیمانه‌بندی و الگوهای طراحی از مهم‌ترین ابزارهای کنترل پیچیدگی هستند که ریشه در تفکر رایانشی دارند. توانایی آزمون‌پذیری و قابلیت اثبات‌پذیری نرم‌افزار نیز در امتداد همین مؤلفه‌ها قرار می‌گیرد.

در سطح سیستم‌های عامل و معماری رایانه نیز تفکر رایانشی حضوری پررنگ دارد. استدلال درباره هم‌روندی، زمان‌بندی پردازش‌ها، اشتراک منابع و بهینه‌سازی کارایی، همگی نمودهای این نوع تفکر هستند. در مقیاس بزرگ‌تر، الگوهای موازی‌سازی همچون پردازش داده‌ای، لوله‌ای و حوزه‌ای، و چارچوب‌هایی نظیر پردازش داده‌های کلان یا مدل تقسیم و ادغام، نمونه‌های بارز تجزیه و ترکیب موازی در حل مسائل پیچیده به شمار می‌روند [۳۴]. این موضوع حتی در آموزش علوم رایانه نیز مورد تأکید قرار گرفته است، چرا که درک معماری‌های موازی و محاسبات با توان بالا بدون برخورداری از مؤلفه‌های تفکر رایانشی دشوار خواهد بود. در زمینه شبکه‌ها و مهندسی اینترنت نیز تفکر رایانشی به صورت طراحی پروتکل‌ها و تحلیل مقیاس‌پذیری و قابلیت اتکاپذیری بروز می‌کند. در حوزه

یه و همکاران (۲۰۲۳) هفت مفهوم کلیدی تفکر رایانشی را که در زمینه ریاضیات و برنامه‌نویسی کاربرد دارند، ارائه داده‌اند. این مفاهیم شامل متغیرها^{۳۰}، شرط‌ها^{۳۱}، حلقه‌ها^{۳۲}، دنباله‌ها^{۳۳}، رویدادها^{۳۴}، زیرروال‌ها^{۳۵} و عملگرها^{۳۶} هستند که همگی در حل مسائل ریاضی و برنامه‌نویسی نقش اساسی دارند. این دیدگاه‌ها، در کنار سایر چارچوب‌های مطرح‌شده توسط برنان و رسنیک (۲۰۱۲)، سلبی و همکاران (۲۰۱۴) و بوکانی و همکاران (۲۰۱۶)، نشان‌دهنده اهمیت و چندگانگی مؤلفه‌های تفکر رایانشی در زمینه‌های مختلف آموزشی، به ویژه در پیوند بین ریاضیات و برنامه‌نویسی هستند. این تنوع در دیدگاه‌ها نشان می‌دهد که تفکر رایانشی یک مفهوم چندوجهی است که از زوایای مختلف قابل بررسی است و مجموعه‌ای از مهارت‌های شناختی را در بر می‌گیرد که برای مواجهه با چالش‌های دنیای امروز ضروری است.

۴- جایگاه تفکر رایانشی در علوم رایانه

تفکر رایانشی در علوم رایانه نه صرفاً به عنوان یک مهارت آموزشی، بلکه به مثابه شیوه‌ای بنیادین از اندیشیدن برای مدل‌سازی، صورت‌بندی و حل مسائل مطرح می‌شود. این مفهوم بر پایه نمایش‌های صوری، تجزیه مسئله، انتزاع، طراحی الگوریتم و ارزیابی کارایی و درستی بنا شده است. یکی از نخستین حوزه‌هایی که جایگاه تفکر رایانشی در آن آشکار می‌شود، منطق و نظریه محاسبه است. در این حوزه، مسئله اصلی شناخت توانایی‌ها و محدودیت‌های محاسبه، بررسی قابلیت تصمیم‌پذیری و تحلیل پیچیدگی زمانی و مکانی الگوریتم‌ها است. ابزارهایی مانند بررسی مدل یا زبان‌های مشخصه‌گذاری صوری همچون TLA⁺، نمونه‌های عینی از به کارگیری تفکر رایانشی در علوم رایانه به شمار می‌روند. در این فرایند، یک مسئله واقعی به مدل‌های حالت محدود کاهش می‌یابد، خواص آن در قالب

- 30-Variables
- 31-Conditionals
- 32-Loops
- 33-Sequences
- 34-Events
- 35-Subroutines
- 36-Operators

سبب می‌شود که حتی پژوهش‌هایی که ریشه آموزشی دارند، همچون آموزش ریاضی، با تکیه بر مفاهیم تفکر رایانشی بتوانند به‌طور طبیعی در چارچوب علوم رایانه قرار گیرند.

۵- نقش تفکر رایانشی در آموزش ریاضی

تفکر رایانشی نه تنها با ریاضیات ارتباط عمیق و تاریخی دارد، بلکه به عنوان ابزاری قدرتمند برای یادگیری، کاوش و کشف مسائل ریاضی عمل می‌کند. زمانی که دانش‌آموزان با اصول تفکر رایانشی به مسائل ریاضی نزدیک می‌شوند، فرایند یادگیری آن‌ها از حالت منفعل به فعال تبدیل می‌شود. این رویکرد به دانش‌آموزان امکان می‌دهد تا مفاهیم انتزاعی ریاضی را تصور و مدل‌سازی کنند، مثلاً با استفاده از برنامه‌نویسی ساده یا نرم‌افزارهای تخصصی می‌توانند نمودارهای توابع، هندسه و پدیده‌های آماری را به صورت بصری بسازند. برای نمونه، طراحی برنامه‌ای برای رسم نمودار یک تابع درجه دوم همان‌طور که در شکل ۲ می‌بینید، این مهارت در مدل‌سازی رایانشی علاوه بر تقویت مهارت‌های کاربردی ریاضی، نشان می‌دهد که چگونه ریاضیات می‌تواند برای حل مسائل دنیای واقعی به کار رود. جنبه دیگر تفکر رایانشی، تشویق به آزمایش و کشف است. به جای حفظ فرمول‌ها، دانش‌آموزان می‌توانند با تغییر پارامترها در برنامه‌ها یا مدل‌ها، نتایج را مشاهده و فرضیه‌های خود را آزمایش کنند. این رویکرد تجربی، کنجکاوی آنان را برمی‌انگیزد و امکان کشف قوانین و روابط ریاضی به‌صورت خودآموز را فراهم می‌کند؛ مانند نوشتن کدهای ساده برای تولید دنباله فیبوناچی و بررسی خواص ناشناخته آن، که درک مفهومی را تقویت کرده و یادگیری را به یک ماجراجویی جذاب تبدیل می‌کند.

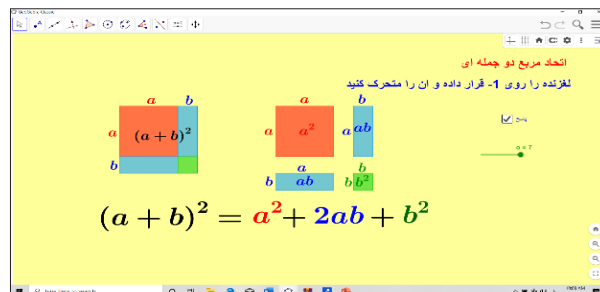
تفکر رایانشی، فراتر از صرف مهارت‌های برنامه‌نویسی است؛ بلکه مجموعه‌ای از فرایندهای فکری است که در فرموله کردن مسائل و بیان راه‌حل‌های آنها به شکلی که بتواند توسط یک عامل پردازش اطلاعات (انسان یا ماشین)

محاسبات ابری، چارچوب مرجع موسسه ملی فناوری و استانداردها^{۳۷} با تعریف پنج ویژگی بنیادین خدمات ابری، مصداق روشنی از انتزاع و خودکارسازی منابع محاسباتی است [۳۵]. در عمل نیز ابزارهایی در این زمینه با بهره‌گیری از همین مؤلفه‌ها، زمان‌بندی خودکار، بهبود خرابی و مدیریت مقیاس‌پذیری در زیرساخت‌های ابری را ممکن می‌سازند [۳۶]. تفکر رایانشی در پایگاه‌های داده و کلان‌داده نیز به روشنی قابل مشاهده است. انتخاب مدل داده‌ای متناسب، طراحی طرح‌واره، ایجاد شاخص‌های کارآمد و برنامه‌ریزی پرس‌وجوها همگی به انتزاع و تحلیل مسئله وابسته‌اند. همان‌گونه که استون‌بریکر و هراشتاین^{۳۸} (۲۰۰۷) بیان کرده‌اند، پایان دوران یک معماری یگانه برای همه، نشان‌دهنده ضرورت مدل‌سازی بارکاری و انتخاب الگوریتم‌ها و معماری‌های داده‌ای متناسب با آن است. این جریان، نمونه‌ای دقیق از تفکر رایانشی در مقیاس داده به شمار می‌رود [۳۷]. در حوزه امنیت داده‌ها و اطلاعات، تفکر رایانشی به‌صورت مدل‌سازی تهدیدها و طراحی نظام‌مند راهکارها تجلی می‌یابد. روش‌های ساخت‌یافته‌ای چون STRIDE، با تجزیه مسئله، انتزاع خصیصه‌های کلیدی و ارزیابی ریسک، فرایندی رایانشی برای درک و کاهش تهدیدات فراهم می‌کنند [۳۸]. در نهایت، در هوش مصنوعی، مؤلفه‌های تفکر رایانشی در قالب جست‌وجو در فضاها، حالت، بازنمایی دانش، بهینه‌سازی و تعمیم‌پذیری ظاهر می‌شوند. الگوریتم‌های جست‌وجو، روش‌های استدلال منطقی و الگوریتم‌های یادگیری ماشین، همگی بر این اساس عمل می‌کنند که یک مسئله واقعی را به نمایش محاسباتی دقیق تبدیل کرده و سپس رویه‌ای مؤثر برای حل آن ارائه دهند [۳۹].

بدین‌ترتیب، می‌توان نتیجه گرفت که تفکر رایانشی در علوم رایانه جایگاهی ریشه‌ای دارد و تقریباً تمامی زیرحوزه‌های این رشته، از نظریه محاسبه گرفته تا هوش مصنوعی و امنیت، بر مؤلفه‌های آن استوارند. این جایگاه

بستری را فراهم می‌کند که دانش‌آموزان می‌توانند مفاهیم انتزاعی ریاضی را به صورت تعاملی و بصری تجربه کنند. ساخت یک بازی که در آن بازیکن باید معادلات ریاضی را حل کند تا امتیاز بگیرد، نه تنها درک معادلات را تقویت می‌کند بلکه مهارت‌های برنامه‌نویسی و تفکر الگوریتمی را نیز توسعه می‌دهد [۴۱].

ارتباط بین تفکر رایانشی و آموزش ریاضی در جنبه‌های گوناگونی قابل مشاهده است که هر یک به نحوی می‌توانند فرایند یاددهی و یادگیری را غنی‌تر سازند. یکی از این جنبه‌ها، نقش تفکر الگوریتمی در حل مسائل ریاضی است؛ جایی که دانش‌آموزان با تبدیل راه‌حل‌های ریاضی به الگوریتم‌های قابل اجرا توسط کامپیوتر، درک عمیق‌تری از منطق نهفته در پس این راه‌حل‌ها پیدا می‌کنند. همچنین، انتزاع به عنوان یکی از مولفه‌های تفکر رایانشی، در فهم مفاهیم انتزاعی ریاضی نقش به‌سزایی دارد، زیرا برنامه‌نویسی امکان ملموس‌سازی این مفاهیم از طریق مدل‌ها و شبیه‌سازی‌ها را فراهم می‌آورد و از طریق شناسایی الگوها و ساخت الگوریتم‌های کلی، به تعمیم مفاهیم ریاضی کمک می‌کند. برنامه‌نویسی نیز به نوبه خود می‌تواند به عنوان یک ابزار قدرتمند برای کاوش و آزمایش ایده‌های ریاضی عمل کند، به طوری که دانش‌آموزان می‌توانند با کدنویسی و مشاهده بصری نتایج تغییرات در پارامترهای مسائل ریاضی، درک شهودی‌تری کسب کنند. در نهایت، در حوزه‌هایی مانند آمار و احتمال برای مثال شبیه‌سازی پرتاب تاس که در شکل ۳ مشاهده می‌کنید باعث می‌شود دانش‌آموزان مفهوم انتزاعی را ملموس‌تر درک کنند و این امر به ویژه برای دانش‌آموزانی با سبک یادگیری بصری اثربخش است. تفکر رایانشی از طریق تجزیه و تحلیل داده‌ها و بصری‌سازی آنها، به فهم عمیق‌تر مفاهیم آماری یاری می‌رساند. این تعامل چندوجهی نشان می‌دهد که چگونه ادغام تفکر رایانشی می‌تواند ابزارهای نوینی برای آموزش و یادگیری موثرتر ریاضیات فراهم آورد.



شکل ۲. نمایش هندسی اتحاد مربع دو جمله‌ای با استفاده از نرم‌افزار جئوجبرا

به طور موثر انجام شود، نقش دارد. این نوع تفکر شامل مولفه‌هایی اساسی نظیر تجزیه (شکستن مسائل پیچیده به اجزای کوچکتر)، تشخیص الگو (یافتن شباهت‌ها و روندها)، انتزاع (تمرکز بر اطلاعات مهم و نادیده گرفتن جزئیات غیرضروری)، و طراحی الگوریتم (ایجاد مجموعه‌ای گام به گام از دستورالعمل‌ها برای حل مسئله) می‌شود [۴۰]. در دنیای امروز که داده‌ها و سیستم‌های پیچیده همه جا حاضرند، تفکر رایانشی به یک شایستگی بنیادین برای درک و تعامل موثر با این پیچیدگی‌ها تبدیل شده است.

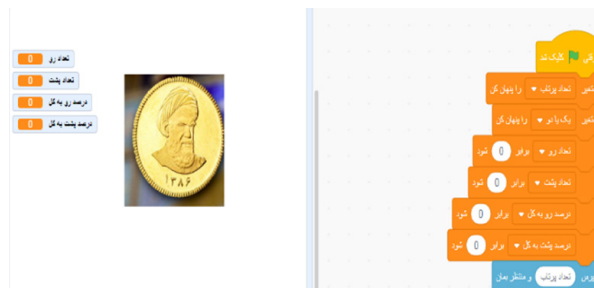
آموزش ریاضی، به قدمت تمدن بشری، همواره به دنبال توسعه توانایی‌های استدلال منطقی، حل مسئله و درک ساختارهای انتزاعی در فراگیران بوده است. هدف از آموزش ریاضی، نه تنها انتقال مجموعه‌ای از فرمول‌ها و رویه‌ها، بلکه پرورش یک ذهن تحلیلی و توانایی مواجهه با چالش‌های کمی در زندگی واقعی است. از طریق آموزش ریاضی، دانش‌آموزان یاد می‌گیرند چگونه مسائل را فرموله کنند، به دنبال الگوها بگردند، استدلال‌های منطقی بسازند و نتایج خود را ارزیابی کنند. این فرایندها، شالوده تفکر انتقادی و توانایی یادگیری مادام‌العمر را تشکیل می‌دهند [۴۲]. اکنون، با در نظر گرفتن ماهیت هر دو حوزه، می‌توان دریافت که تفکر رایانشی و آموزش ریاضی در بسیاری از جنبه‌ها همپوشانی دارند و می‌توانند به طور متقابل یکدیگر را تقویت کنند. تفکر رایانشی، با تأکید بر رویکردهای ساختاریافته برای حل مسئله، می‌تواند ابزارهای قدرتمندی را در اختیار آموزش ریاضی قرار دهد. به عنوان مثال، برنامه‌نویسی، به ویژه برنامه‌نویسی بلوکی مانند اسکرچ،

بنیان‌های اصیل علوم رایانه استوار باشد. پرورش تفکر رایانشی در بستر آموزش ریاضی می‌تواند ضمن ارتقای فهم عمیق‌تر مفاهیم ریاضی، به توسعه توانایی‌های تحلیلی، حل مسئله، خلاقیت و تفکر انتقادی دانش‌آموزان نیز منجر شود. چنین رویکردی در نهایت زمینه‌ساز تربیت نسلی خواهد بود که قادر است با چالش‌های پیچیده‌ی قرن بیست‌ویکم در حوزه‌های علمی، فناوریانه و اجتماعی مواجه شود. پژوهش‌ها نیز بر این موضوع صحه گذاشته‌اند که پرورش مهارت‌های تفکر رایانشی نه تنها توانمندی‌های ریاضی را افزایش می‌دهد، بلکه بر خلاقیت، منطق، و تفکر انتقادی نیز تأثیر مثبت دارد [۱۸، ۱]، همچنین شواهد موجود نشان می‌دهد که به‌کارگیری محیط‌هایی چون اسکرچ، جتو جبرا یا پایتون در آموزش ریاضی می‌تواند بستری ملموس برای به‌کارگیری مفاهیم انتزاعی ریاضی فراهم آورد و فهم دانش‌آموزان را عمیق‌تر کند.

از منظر آینده‌پژوهی، مسیرهای متعددی برای پژوهش‌های بعدی گشوده می‌شود. بررسی نحوه ادغام چارچوب‌های نظری علوم رایانه در طراحی فعالیت‌های آموزشی ریاضی، تحلیل اثرات میان‌مدت و بلندمدت آموزش تفکر رایانشی بر یادگیری دانش‌آموزان، و توسعه ابزارهای فناوریانه نوین برای پشتیبانی از این فرایند، از جمله زمینه‌های مهمی هستند که نیازمند پژوهش‌های عمیق‌تر می‌باشند. در مجموع، می‌توان نتیجه گرفت که تفکر رایانشی نه صرفاً ابزاری آموزشی یا مهارتی فردی، بلکه رویکردی راهبردی و میان‌رشته‌ای است که ریشه در علوم رایانه دارد و می‌تواند نقشی کلیدی در تحول آموزش ریاضی و آماده‌سازی نسل آینده برای جامعه دیجیتال ایفا کند.

مراجع

1. Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35. <https://doi.org/10.1145/1118178.1118215>
2. Wing, J. (2011). Research notebook: Computational thinking—What and why. The link magazine, 6, 20-23. <https://people.cs.vt.edu/~kafura/CS6604/Papers/CT->



شکل ۳. شبیه‌سازی پرتاب سکه برای محاسبه احتمال در برنامه نویسی اسکرچ

۶- بحث و نتیجه‌گیری

تفکر رایانشی در دو دهه اخیر به‌عنوان یکی از مفاهیم بنیادین علوم رایانه جایگاهی استوار یافته است. بررسی ادبیات نشان داد که این مفهوم نه تنها در حوزه‌های نظری چون منطق، نظریه محاسبه و تحلیل الگوریتم‌ها، بلکه در حوزه‌های کاربردی نظیر مهندسی نرم‌افزار، محاسبات موازی، کلان‌داده، امنیت اطلاعات و هوش مصنوعی نیز حضوری پررنگ دارد [۳۹، ۹]. مؤلفه‌هایی همچون تجزیه مسئله، تشخیص الگو، انتزاع، طراحی الگوریتم و تعمیم‌پذیری، شالوده بسیاری از پیشرفت‌های علمی و فناوریانه در علوم رایانه را تشکیل داده‌اند و نقش آنها در توسعه سامانه‌های محاسباتی امروزی غیرقابل انکار است. با این حال، مرور کارهای مرتبط نشان داد که علی‌رغم اهمیت فزاینده تفکر رایانشی در علوم رایانه، هنوز شکاف معناداری میان بنیان‌های نظری این حوزه و کاربردهای آموزشی آن به‌ویژه در آموزش ریاضی وجود دارد. پژوهش‌های آموزشی بیشتر بر ابزارهایی نظیر برنامه‌نویسی اسکرچ و لوگو یا بر اثرات کلی آموزش تفکر رایانشی متمرکز بوده‌اند [۱۵، ۶]، در حالی که پیوند عمیق با مفاهیم فنی علوم رایانه همچنان کمتر مورد توجه قرار گرفته است.

بر این اساس، مقاله حاضر بر ضرورت بازاندیشی در آموزش ریاضی تأکید دارد؛ بازاندیشی‌ای که نه تنها به تلفیق سطحی فناوری‌های نوین بسنده نکند، بلکه بر

- ing in K-12: A Review of the State of the Field. *Educational Researcher*, 42(1), 38-43. <https://doi.org/10.3102/0013189X12463051>
19. Papert, S., & Harel, I. (1991). *Situando o construcionismo*. Norwood, NJ: Ablex.
20. Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association (AERA)*, Vancouver, Canada.
21. Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in human behavior*, 72, 678-691. <https://doi.org/10.1016/j.chb.2016.08.047>
22. Becker, B. A., Bradley, S., Maguire, J., Black, M., Crick, T., Saqr, M., ... & Quille, K. (2023). Computing Education Research in the UK & Ireland. In *Past, Present and Future of Computing Education Research: A Global Perspective* (pp. 421-479).
23. Lee, I., Grover, S., Martin, F., Pillai, S., & Malyn-Smith, J. (2020). Computational thinking from a disciplinary perspective: Integrating computational thinking in K-12 science, technology, engineering, and mathematics education. *Journal of Science Education and Technology*, 29, 1-8.
24. Montiel, H., & Gomez-Zermeño, M. G. (2021). Educational Challenges for Computational Thinking in K-12 Education: A Systematic Literature Review of "Scratch" as an Innovative Programming Tool. *Computers*, 10(6), 69. <https://doi.org/10.3390/computers10060069>
25. Ye, H., Liang, B., Ng, O., & Chai, C. S. (2023). Integration of computational thinking in K-12 mathematics education: a systematic review on CT-based mathematics instruction and student learning. *International Journal of STEM Education*, 10(1), 1-26. <https://doi.org/10.1186/s40594-023-00396-w>
26. Hartley, R., & Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge university press.
27. Robert, C. (2014). *Machine learning, a probabilistic perspective*.
28. Segal, H., & Kalfon- Hakhmigari, M. (2025). Grit as a moderator of the association between cognitive abilities and STEM achievements in high school. *IJ STEM Ed* 12, 25.
29. Futschek, G. (2006, November). Algorithmic thinking: the key for understanding computer science. In *International conference on informatics in secondary schools-evolution and perspectives* (pp. 159-168). Berlin, Heidelberg: Springer Berlin Heidelberg.
30. Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational research review*, 22, 142-158.
31. Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning* (Vol. 1, No. 2). Cambridge: MIT
- What-And-Why.pdf
3. Denning, P. J. (2009). The profession of IT Beyond computational thinking. *Communications of the ACM*, 52(6), 28-30.
4. Papert, S. (1980). *Children, computers, and powerful ideas* (Vol. 10, pp. 978-3). Eugene, OR, USA: Harvester. Commission on Mathematics Instruction international. [https://doi.org/10.1016/0732-118x\(83\)90034-x](https://doi.org/10.1016/0732-118x(83)90034-x)
5. Papert, S. (2000). What's the big idea? Toward a pedagogy of idea power. *IBM systems journal*, 39(3.4), 720-729. <https://doi.org/10.1147/sj.393.0720>
6. Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, 52(11), 60-67. <https://doi.org/10.1145/1592761.1592779>
7. Weintrop, D., Beheshti, E., Horn, M. S., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25(1), 127-147.
8. Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM*, 60(6), 33-39.
9. Denning, P. J., & Tedre, M. (2021). Computational thinking: A disciplinary perspective. *Informatics in Education*, 20(3), 361.
10. Grumberg, O., Clarke, E. M., & Peled, D. (1999). *Model checking*. In *International Conference on Foundations of Software Technology and Theoretical Computer Science*; Springer: Berlin/Heidelberg, Germany.
11. Cormen, T., Leiserson, C., Rivest, R., & Stein, C. (2009). *Book: introduction to algorithms*. Algorithms, 1(1L9K1), 1L9K1.
12. Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1), 107-113.
13. Hellerstein, J. M., Stonebraker, M., & Hamilton, J. (2007). Architecture of a database system. *Foundations and Trends® in Databases*, 1(2), 141-259.
14. Nordby, S. K., Bjerke, A. H., & Mifsud, L. (2022). Computational thinking in the primary mathematics classroom: A systematic review. *Digital Experiences in Mathematics Education*, 8(1), 27-49.
15. Yadav, A., Stephenson, C., & Hong, H. (2017). Computational thinking for teacher education. *Communications of the ACM*, 60(4), 55-62.
16. Selby, C. C. (2014). *How can the teaching of programming be used to enhance computational thinking skills?* (Doctoral dissertation, University of Southampton).
17. Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM inroads*, 2(1), 48-54.
18. Grover, S., & Pea, R. (2013). *Computational Think-*

- press.
32. Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., & Punie, Y. (2016). Developing computational thinking in compulsory education. European Commission, JRC Science for Policy Report, 68. <https://data.europa.eu/doi/10.2791/792158>
 33. Lampert, L. (2002). Specifying systems (Vol. 388). Boston: Addison-Wesley.
 34. Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1), 107-113.
 35. Mell, P., & Grance, T. (2011). The NIST definition of cloud computing.
 36. Burns, B., Grant, B., Oppenheimer, D., Brewer, E., & Wilkes, J. (2016). Borg, omega, and kubernetes. *Communications of the ACM*, 59(5), 50-57.
 37. Hellerstein, J. M., Stonebraker, M., & Hamilton, J. (2007). Architecture of a database system. *Foundations and Trends® in Databases*, 1(2), 141-259.
 38. Shostack, A. (2014). Threat modeling: Designing for security. John Wiley & sons.
 39. Russell, S., Norvig, P., Popineau, F., Miclet, L., & Cadet, C. (2021). *Intelligence artificielle: une approche moderne* (4^e édition). Pearson France.
 40. Selby, C., & Woollard, J. (2013). Computational thinking: the developing definition.
 41. Jackiw, N., & Sinclair, N. (2010). Learning through teaching, when teaching machines: Discursive interaction design in sketchpad. In *Learning Through Teaching Mathematics: Development of Teachers' Knowledge and Expertise in Practice* (pp. 153-168). Dordrecht: Springer Netherlands.
۴۲. گویا، زهرا و آشنا، امیرحسین (۱۳۹۵). از آموزش و ریاضی تا آموزش ریاضی. *فرهنگ و اندیشه ریاضی*، ۳۵(۱)، ۹-۵۱.