

روش‌های نمایش جواب

الگوریتم‌های فرا ابتکاری

حسین کریمی

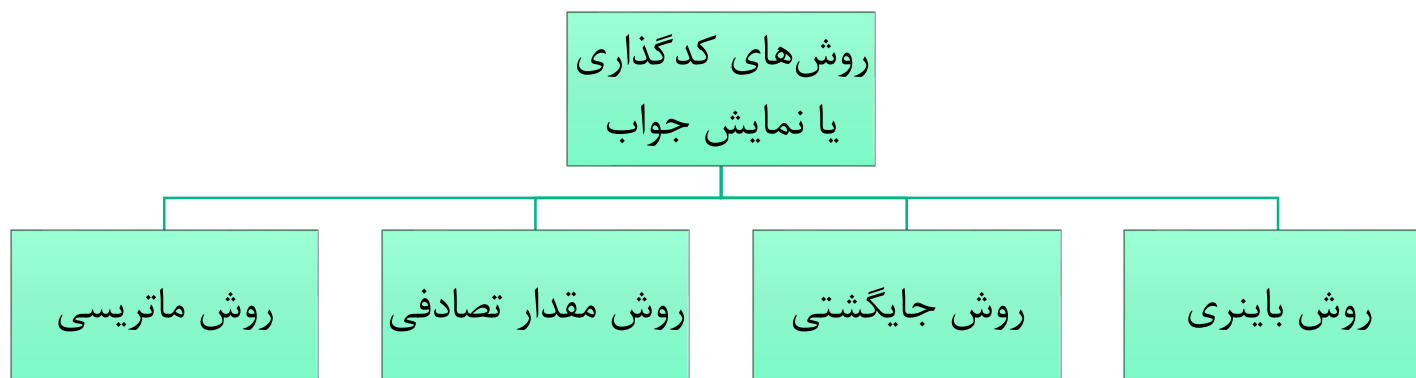
- مقدمه
- کدگذاری باینری
- کدگذاری جایگشتی
- کدگذاری مقدار تصادفی
- کدگذاری ماتریسی



- در این قسمت به بررسی کامل انواع کدگذاری خواهیم پرداخت.
- الگوریتم‌های فراابتکاری به جای این که بر روی پارامترها یا متغیرهای مسئله کار کند، با شکل کد شده آنها سر و کار دارند.
- تعداد بیت‌هایی که برای کدگذاری متغیرها استفاده می‌شود، به دقت مورد نظر برای جواب‌ها، محدوده تغییر پارامترها و رابطه بین متغیرها وابسته است.
- رشته یا دنباله‌ای از بیت‌ها که به عنوان شکل کد شده یک جواب از مسئله مورد نظر می‌باشد، یک جواب نامیده می‌شود.



- انواع روش‌های کدگذاری برای مسائل تصمیم‌گیری به صورت زیر است.





- کد گذاری باینری ساده‌ترین کدگذاری است اما در مسائل پیچیده این نوع تبدیل چندان مناسب نیست؛ چون معمولاً باعث می‌شود طول کروموزوم‌ها برای نگهداری اطلاعات پاسخ، بسیار بزرگ شود.
- در تبدیل باینری، اعضای جمعیت به رشته‌هایی از صفرها و یک‌ها تبدیل می‌شوند.



■ مثال

- فرض کنید الگوریتم می‌خواهد ماکزیمم تابع $F(x,y,z)$ را پیدا کند. در نظر بگیرید جستجو باید در اعداد صحیح مثبت و در محدوده ۰ تا ۲۵۵ انجام شود هر پاسخ ممکن شامل سه عدد X و Y و Z می‌باشد.
- طول هر عدد در محدوده مورد نظر مسئله در تبدیل باینری حداکثر ۸ بیت می‌باشد. اگر هر کروموزوم را به صورت XYZ در نظر بگیریم بنابراین برای پوشش دادن به تمام پاسخ‌های ممکن لازم است طول کروموزوم ۲۴ بیت باشد. برای این مسئله کروموزوم C می‌تواند به شکل زیر باشد.

$C=11010010 \ 11100011 \ 00110111$

کدگذاری باینری – ادامه

- در همین مثال اگر لازم باشد اعداد منفی نیز جستجو شود می توان یک بیت به ابتدای هر رشته اضافه کرد که مثلاً اگر ۰ باشد، عدد، مثبت و اگر ۱ باشد عدد، منفی در نظر گرفته شود.

$$000000001=1$$

$$100000001=-1$$

- تبدیل اعداد اعشاری نیز می تواند با استفاده از چنین تمهیداتی انجام شود.

■ مثالی دیگر

- فرض کنید که می‌خواهید مسئله کوله‌پشتی را حل کنید.
- فرض کنید تعدادی گزینه در اختیار داریم که باید زیرمجموعه‌ای از آنها انتخاب شود تا کوله‌پشتی با بیشترین ارزش پر شود ولی کوله‌پشتی فضای محدودی دارد. هر کدام از گزینه‌ها دارای ارزش و وزنی هستند. این مسئله را می‌توان با کدگذاری باینری نوشت.
- به عنوان مثال، اگر ۸ گزینه وجود داشته باشد. کد باینری 01011010 یک پاسخ برای این مسئله است. یعنی گزینه‌های ۲، ۴، ۵، ۷ انتخاب می‌شوند. البته لازم به ذکر است که ممکن است این پاسخ نشدنی باشد که با حذف تصادفی گزینه‌های انتخابی تا جایی که تمامی گزینه‌های در کوله جای بگیرند، تعمیر می‌شود.

روش کدگذاری جایگشتی



- در این روش، جواب‌ها به صورت رشته‌ای از اعداد طبیعی نشان داده می‌شوند که هرکدام از این اعداد، مربوط به پارامتر ویژه‌ای در فضای حل مسئله است.
- ترتیب قرارگیری این اعداد مهم بوده و طول رشته دقیقاً با تعداد پارامترهای تعریف شده در مسئله برابر است.
- کاربرد معروف این نوع کدگذاری در حل مسئله فروشنده دوره‌گرد است.
- البته این روش از پرکاربردترین روش‌های کدگذاری برای مسائل تصمیم‌گیری است.



روش کدگذاری جایگشتی-ادامه

- در بسیاری از مسایل مانند مسئله «فروشنده دوره گرد» با جایگشت‌های مختلفی از مجموعه راه‌حل‌ها رو به رو هستیم.
- در این مسئله تعدادی شهر داریم که فاصله میان آنها معلوم است و با شروع از یک شهر و ختم به همان شهر می‌بایست؛ از تمام شهرها فقط و فقط یکبار عبور نمائیم و کمترین مسافت ممکن طی شود.
- به عنوان مثال، اگر ۶ شهر وجود داشته باشد، تعداد حالات کلی جهت ایجاد یک دور همیلتونی که یک پاسخ برای مسئله است برابر ۶! خواهد بود (البته برخی حالات مشابه هم هستند).

روش کدگذاری جایگشتی-ادامه

- به عنوان مثال جایگشتی که در ادامه نشان داده شده است، یک جواب برای مسئله فروشنده دوره‌گرد خواهد بود که به ترتیب شهرهای اول، پنجم، سوم، چهارم، ششم و دوم را طی خواهد کرد.

| | | | | | | |
|-------------|---|---|---|---|---|---|
| ترتیب شهرها | ۱ | ۵ | ۳ | ۴ | ۶ | ۲ |
|-------------|---|---|---|---|---|---|

روش کدگذاری جایگشتی-ادامه

- از روش جایگشتی، گاهی اوقات به صورت جایگشت گسسته از هم استفاده می‌شود.
- به عنوان مثال، فرض کنید که مسئله چند فروشنده دوره گرد در یک مبدأ وجود داشته باشد. یعنی به عنوان مثال، سه فروشنده دوره گرد می‌تواند استفاده شود. در این صورت برای مثال قبل، از جایگشتی گسسته استفاده می‌شود، معمولاً چند نمایش گسسته در جایگشت، از عدد صفر استفاده می‌شود.

روش کدگذاری جایگشتی-ادامه

- جایگشت نمونه بیان شده به این صورت است که اولین فروشنده، ابتدا از مبدأ صفر به شهر اول و بعد به شهر پنجم و سپس به همان مبدأ باز می‌گردد. فروشنده دوم اصلاً مسیری برای حرکت ندارد و فروشنده سوم به ترتیب از مبدأ به شهرهای سوم، چهارم، ششم و دوم می‌رود، و در پایان به مبدأ باز می‌گردد.

| | | | | | | | | |
|-------------|---|---|---|---|---|---|---|---|
| ترتیب شهرها | ۱ | ۵ | ۰ | ۰ | ۳ | ۴ | ۶ | ۲ |
|-------------|---|---|---|---|---|---|---|---|

- مسائل دیگری همانند زمان‌بندی پروژه‌ها، زمان‌بندی جریان کارگاهی، مسیریابی وسایل نقلیه، کوتاه‌ترین مسیر و غیره هم می‌توانند از طریق کدگذاری جایگشتی حل شوند.

روش کدگذاری مقدار تصادفی

- در این نوع روش کدگذاری که به Random Key نیز مشهور است، جواب‌ها می‌توانند هر نوع داده مرتبط با مسئله را در اختیار نمایند.
- این داده‌ها می‌توانند از نوع اعداد حقیقی، عبارات منطقی، دستورات جهت‌یابی، داده‌های کد شده به صورت رشته‌های حرفی و غیره باشند.
- اغلب روش‌های فراابتکاری به خصوص الگوریتم‌های تکاملی همانند، ژنتیک، تجمع ذرات و غیره در فضای حل پیوسته نوشته شده‌اند لذا در مسائلی که متغیرها به صورت عدد صحیح هستند، تبدیل فضای پیوسته به گسسته نیاز است.

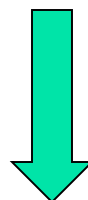
روش کدگذاری مقدار تصادفی – ادامه

- یکی از کاربردهای روش کدگذاری مقدار در همین موضوع است. اگر به جاری بردار جواب a برای یک جواب شدنی از بردار زیر استفاده شود، آنگاه با تمامی این اعداد می‌شود در فضای الگوریتم‌های تکاملی کار کرد.
- در فرمول‌های الگوریتم‌های فراابتکاری از اعداد تصادفی بالا استفاده می‌شود. اما برای تبدیل آن‌ها به یک جایگشت تصادفی، ابتدا باید این اعداد را به صورت صعودی یا نزولی مرتب‌سازی کرد.

روش کدگذاری مقدار تصادفی – ادامه

■ مثال

| | | | | | | | | | | |
|------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| عدد تصادفی | ۰.۰۶۱ | ۰.۱۶۱ | ۰.۳۵۸ | ۰.۰۷۶ | ۰.۲۴۹ | ۰.۵۳۵ | ۰.۳۵۴ | ۰.۹۲۳ | ۰.۴۸۹ | ۰.۲۰۲ |
| جایگاه عدد | ۱ | ۲ | ۳ | ۴ | ۵ | ۶ | ۷ | ۸ | ۹ | ۱۰ |



| | | | | | | | | | | |
|------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| عدد تصادفی | ۰.۰۶۱ | ۰.۰۷۶ | ۰.۱۶۱ | ۰.۲۰۲ | ۰.۲۴۹ | ۰.۳۵۴ | ۰.۳۵۸ | ۰.۴۸۹ | ۰.۵۳۵ | ۰.۹۲۳ |
| جایگاه عدد | ۱ | ۴ | ۲ | ۱۰ | ۵ | ۷ | ۳ | ۹ | ۶ | ۸ |

روش کدگذاری مقدار تصادفی – ادامه

- جایگاه اعداد پس از مرتب سازی، همان جایگشت تصادفی خواهد بود که به دنبال آن هستید.
- با این ترتیب اعداد بدست آمده می‌توانید همانند بردار تصادفی a برخورد کنید.
- به این روش کدینگ الگوریتم‌ها، روش اعداد تصادفی می‌گویند.
- از این روش نمایش جواب در الگوریتم‌هایی که منطق و فرمول‌هایی مطابق فضای پیوسته دارند، می‌شود استفاده کرد.

روش کدگذاری ماتریسی

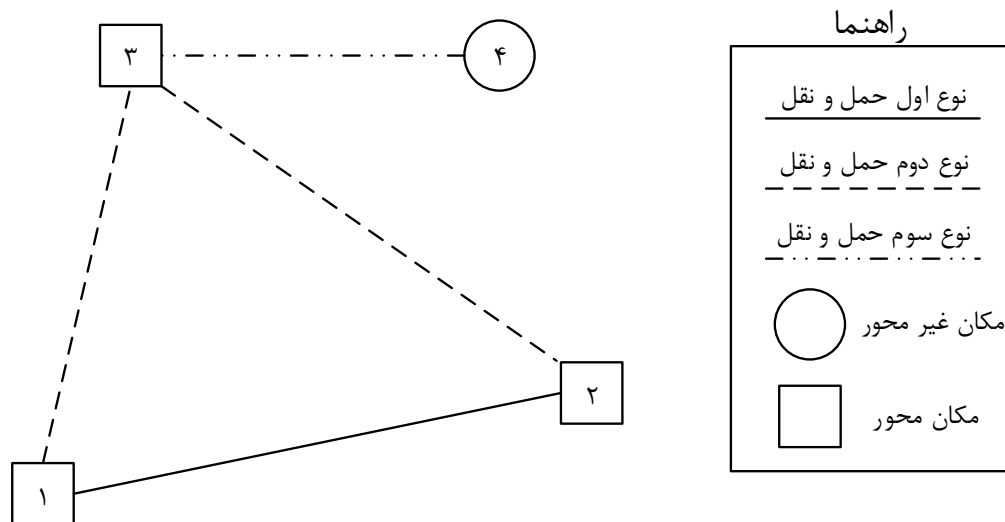


- این روش کدگذاری در مسائل مکان‌یابی و همچنین در مسائل زمان-بندی بسیار پرکاربرد است.
- در واقع هنگامی که نشود از رویکرد برداری همانند رویکرد جایگشتی استفاده کرد و تنوع متغیرهای تصمیم و ابعاد آن بزرگ باشد، آن‌گاه از رویکرد کدگذاری ماتریسی استفاده می‌شود.



روش کدگذاری ماتریسی – ادامه

- به عنوان مثال فرض کنید که یک جواب مسئله مکان‌یابی محور که در ادامه آورده شده است، نمایش داده شده است.
- در این شبکه، برای هر یال قرار است که نوع حمل و نقل نیز تصمیم‌گیری شود. به همین دلیل به راحتی نمی‌توان در یک بردار این شبکه را کدگذاری کرد.



روش کدگذاری ماتریسی – ادامه

- برای این شبکه، ماتریس زیر که دارای سه ردیف و چهار ستون است؛ به ترتیب شماره مکان، نوع تخصیص و نوع حمل و نقل را نشان می‌دهد. البته شماره مکان که در طول الگوریتم تغییری نخواهد کرد و تنها برای درک بهتر نمایش جواب، نشان داده شده است.
- این ماتریس فقط ارتباط بین غیرمحورها با محورها را نشان می‌دهد. البته اگر مکانی محور باشد لزوماً باید به خودش تخصیص یابد و نوع حمل و نقلی برای آن وجود ندارد. به همین دلیل از "-" استفاده شده است.

| شماره مکان | ۱ | ۲ | ۳ | ۴ |
|---------------|---|---|---|---|
| تخصیص | ۱ | ۲ | ۳ | ۳ |
| نوع حمل و نقل | - | - | - | ۳ |

روش کدگذاری ماتریسی – ادامه

- ماتریسی که در ادامه آورده شده است، ارتباط بین محورها را نشان می‌دهد. چون ارتباط بین محورها نوعی ارتباط دو طرفه است، لزوماً باید از کدگذاری ماتریسی استفاده شود. چون در مسئله مکان‌یابی محور کلاسیک، تمامی محورها با هم در ارتباط هستند، به همین دلیل تمامی درآیه‌ها به جز قطر اصلی عدد خواهند گرفت و نوع حمل و نقلی بین دو محور را نشان می‌دهد. این مثال به خوبی نحوه تعریف کدگذاری ماتریسی را نشان می‌دهد. لازم به ذکر است که تا جایی که امکان دارد، ابتدا از روش‌های برداری، به خصوص جایگشتی استفاده شود و در صورت نیاز، باید کدگذاری ماتریسی نوشته شود.

| محور محور | ۱ | ۲ | ۳ |
|--------------|---|---|---|
| ۱ | - | ۱ | ۲ |
| ۲ | ۱ | - | ۲ |
| ۳ | ۲ | ۲ | - |