



نیمسال اول ۱۴۰۰-۱۴۰۱

مدرس: دکتر مجتبی رفیعی

ساختمان داده‌ها و الگوریتم‌ها

جلسه ۶

نگارنده: سیده طوبی حسینی

۲۲ مهر ۱۴۰۰

فهرست مطالب

- ۱ الگوریتم تقریبی (Approximation algorithm)
- ۲ الگوریتم تصادفی (Randomized algorithm)

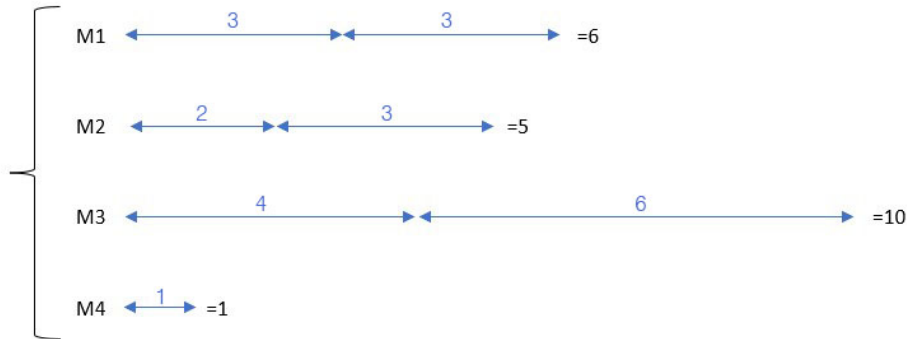
۱ الگوریتم تقریبی (Approximation algorithm)

الگوریتم‌های تقریبی، رویکردی برای حل مسئله به منظور دستیابی به جوابی نزدیک جواب اصلی و نه جواب دقیق مسئله است. چنین رویکردی برای غلبه بر مشکلاتی همچون عدم کارایی حل مسئله اصلی به سبب دشواری و زمانبر بودن حل مساله اصلی ارایه شده است. برای فهم بهتر رویکرد الگوریتم‌های تقریبی، مثالی در ادامه آورده شده است.

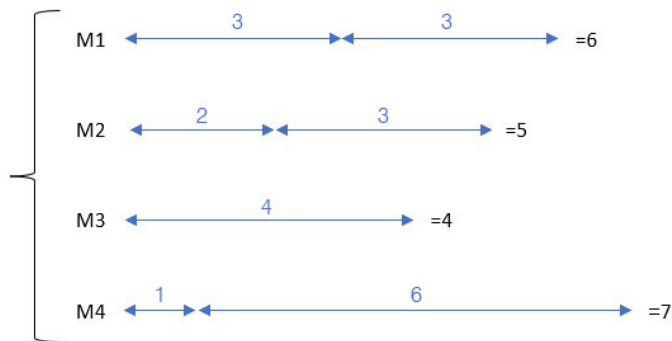
عنوان مسئله: تخصیص پردازش
شرح مسئله: تقسیم n پردازش روی m ماشین به نحوی که زمان پردازش کلی همه آنها کمینه شود.

process: 3 , 2 , 4 , 1 , 3 , 3 , 6
Machine: M1 , M2 , M3 , M4

اگر بخواهیم ۷ پردازش با واحدهای زمانی مشخص را بین ۴ ماشین تقسیم کنیم، اولین و ساده‌ترین راهی که به ذهن می‌رسد این است که پردازش‌های ۱ تا ۴ را به ماشین‌های ۱ تا ۴ و سپس سه پردازش باقی مانده را به ماشین‌های اول تا سوم اختصاص دهیم.



همانطور که مشاهده می‌شود روش فوق به ۱۰ واحد زمانی برای پردازش همه داده‌ها نیاز دارد که به نظر می‌رسد روش کارآمدی نیست و می‌توان در مدت زمان کمتری نیز این پردازش را انجام داد. به عنوان مثال می‌توان پردازشی که ۶ واحد زمانی به طول می‌انجامد را به جای ماشین ۳ به ماشین ۴ اختصاص داد و مدت زمان پردازش نهایی را به اندازه ۳ واحد زمانی کاهش داد و به ۷ رساند:

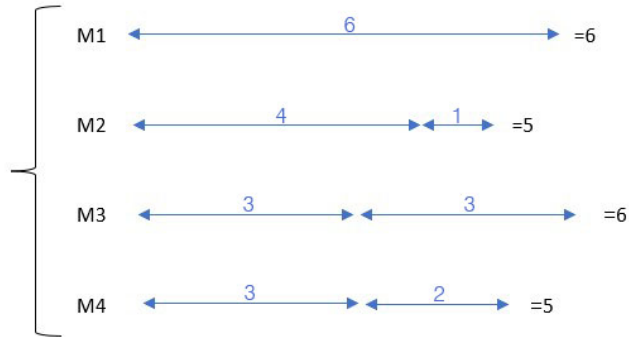


رویکرد دیگری که برای حل این مسئله وجود دارد، رویکرد حریصانه (Greedy) نام دارد که طی ۳ مرحله و به ترتیب زیر انجام می‌شود:

- مرتب کردن پردازش‌ها به صورت نزولی،
 - تخصیص به ترتیب پردازش‌ها به هر یک از ماشین‌ها،
 - پیدا کردن ماشین با کمترین زمان اجرا و تخصیص پردازش‌های باقی‌مانده به طور نزولی.
- پس برای حل مسئله فوق ابتدا پردازش‌های داده شده را به صورت نزولی مرتب می‌کنیم:

process: 6 , 4 , 3 , 3 , 3 , 2 , 1

سپس به ترتیب گفته شده و مطابق شکل زیر هر پردازش را به هر یک از ماشین‌ها اختصاص می‌دهیم.

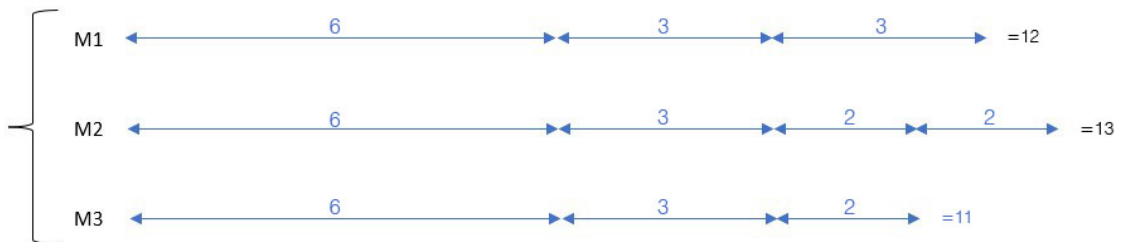


مشاهده می‌شود که با استفاده از روش حریصانه توانستیم که در ۶ واحد زمانی پردازش را به اتمام برسانیم که نسبت به روش‌های قبلی بسیار مطلوب است.
نکته جالب توجه آن است که برخلاف انتظار، رویکرد حریصانه همواره جوابگو نیست! در این رابطه به مثال زیر دقت کنید.

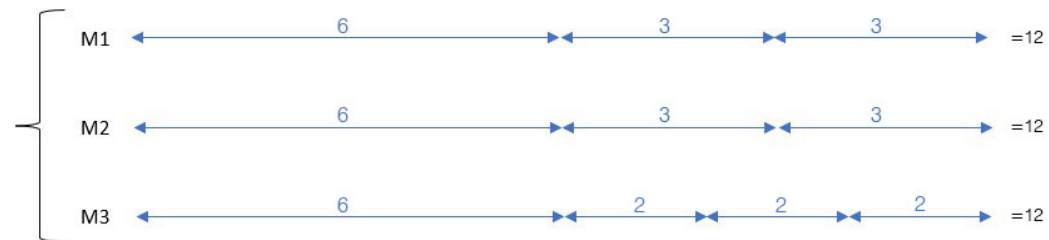
process: 6 , 6 , 6 , 3 , 3 , 3 , 3 . 2 . 2 , 2

Machine: M1 , M2 , M3

تخصیص پردازش‌ها طبق رویکرد حریصانه:



تخصیص پردازش‌ها به گونه‌ای که از رویکرد حریصانه بهینه تر است:



با مقایسه‌ی دو نمودار فوق و با توجه به اینکه حل این سوال طبق رویکرد حریصانه، یک واحد زمانی بیشتر نیاز دارد متوجه می‌شویم که همواره این رویکرد کاراترین روش برای تخصیص پردازش‌ها نخواهد بود.

۲ الگوریتم تصادفی (Randomized algorithm)

به الگوریتم‌هایی اطلاق می‌شود که از مقادیر تصادفی برای حل مسئله استفاده می‌کنند. افزایش کارایی و اهداف امنیتی را می‌توان دو نمونه از کاربردهای اصلی چنین رویکردهای الگوریتمی دانست.

الگوریتم‌های تصادفی را می‌توان بر اساس میزان دقت در خروجی به دو رده کلی زیر تقسیم‌بندی کرد:

- همواره جواب صحیح برمی‌گردانند.

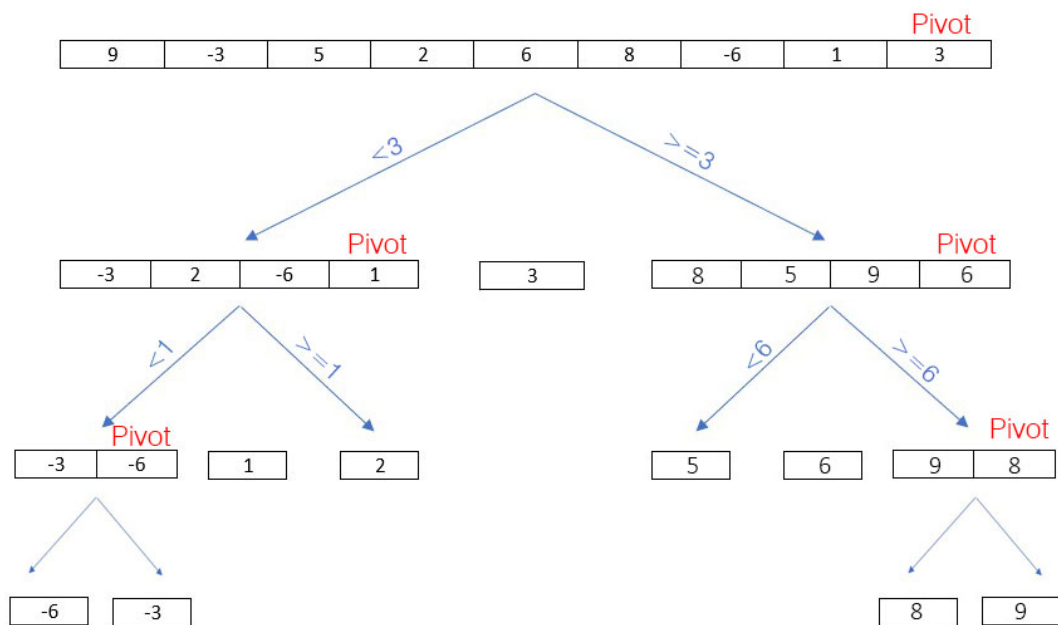
- با یک احتمال مشخص جواب درست می‌دهند و ممکن است جواب اشتباه نیز تولید کنند.

برای فهم بهتر رویکرد الگوریتم‌های تصادفی، مثالی در ادامه آورده شده است.

عنوان مسئله: مرتب سازی یک لیست از اعداد

راه حل مسئله: مرتب سازی سریع (quick sort)

در نمودار زیر نحوه مرتب سازی ۹ عدد دلخواه در ۳ مرحله و با استفاده از الگوریتم مرتب سازی سریع نشان داده شده است به گونه‌ای که در هر مرحله یکی از عناصر آرایه را به عنوان اهرم pivot قرار می‌دهد و سپس سایر عناصر آرایه را نسبت به این عنصر می‌سنجد و اعداد بزرگ‌تر را بعد از این عنصر و اعداد کوچک‌تر را قبل از این عنصر قرار می‌دهد و تا جایی که کل آرایه مرتب شود، این عمل را تکرار می‌کند.



همانطور که در نمودار فوق مشاهده می‌شود، آرایه‌های تک عنصری به ترتیب از سمت چپ به راست، مرتب شده‌ی آرایه‌ی اولیه را نشان می‌دهند.