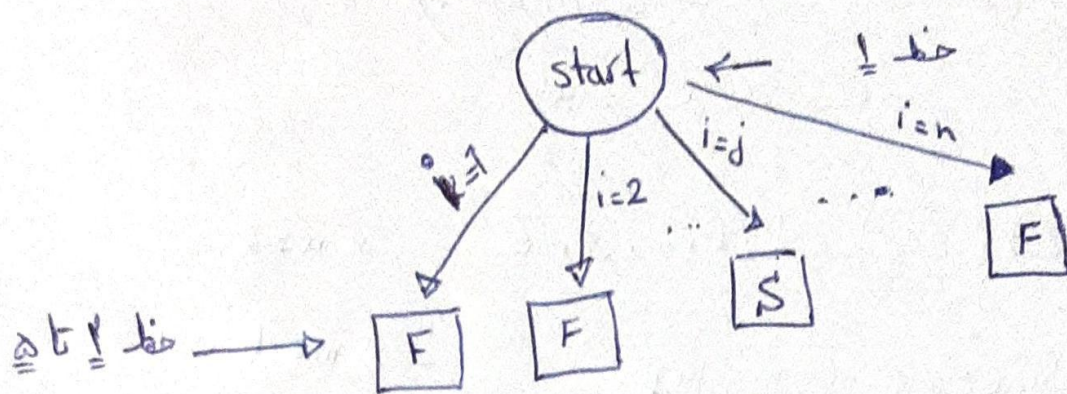


۱۳) با فرض اینکه $K = |A|$ است، نحوه بارش آلو ریتیم NDsearch به صورت زیر است:



۱۴) آلو ریتیم های تقریبی (Approximation algorithms):

اقدام به حل مسأله به منظور دستیابی به یک جواب نزدیک جواب اصلی و نه دقیقاً جواب اصلی ← مثلاً علت می توان این باشد که حل مسأله برار جواب دقیق کارا نیست
امای تران به طور کارا بر جواب تقریبی بدست آورد.

۱۵) مثال بار آلو ریتیم های تقریبی:

Makespan Scheduling Problem

← تحقیق پردازش

صورت مسأله: تقسیم n پردازش روی m ماشین به نحوی که زمان کلی پردازش همه آنها کمینه شود.

Process: 3, 2, 4, 1, 3, 3, 6 → $n = 7$

Machine: m_1, m_2, m_3, m_4 → $m = 4$

m_1 | 3 | 3 |

m_2 | 2 | 3 |

m_3 | 4 | 6 |

m_4 | 1 |

را دهنده پردازش ها را به ترتیب بین ماشین ها تقسیم کنیم

زمان مورد نیاز برای پردازش = 10

۵

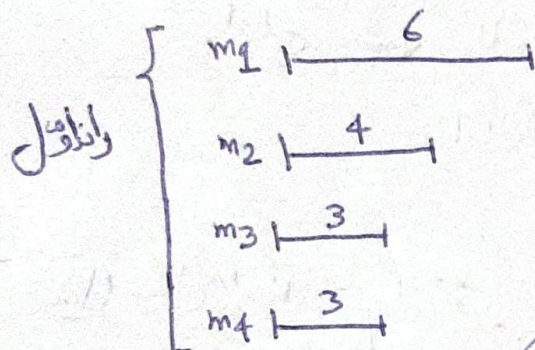
راه حل ۲: روش درجه بندی
 مرتب کردن پردازش‌ها به صورت نزولی
 راندها: تقسیم به ترتیب به ماشین‌ها
 راندهای بعدی: پیک کردن ماشین با کمترین زمان اجرا و تقسیم پردازش‌ها باقی مانده
 به حسب: بر اساس ترتیب نزولی

2-Approximation

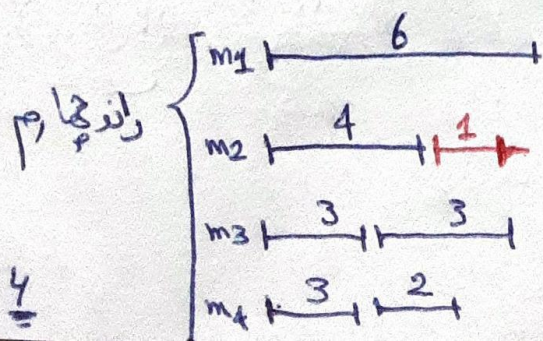
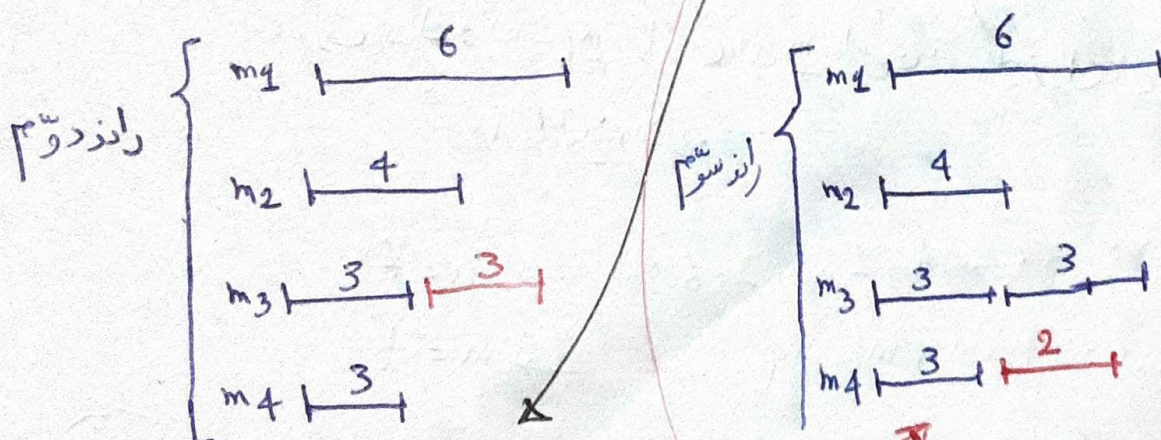
Process: 3, 2, 4, 1, 3, 3, 6 $\rightarrow n=6$

Machine: $m_1, m_2, m_3, m_4 \rightarrow m=4$

مرتب سازی نزولی پردازش‌ها $\rightarrow 6, 4, 3, 3, 3, 2, 1$



مرحله دوم درجه بندی
 Process: 6, 6, 6, 3, 3, 3, 3, 2, 2, 2
 Machine: m_1, m_2, m_3



زمان مورد نیاز برای پردازش = 9
 در راه حل دوم

⑫ الگوریتم‌های تصادفی (Randomized Algorithms):

به الگوریتم‌هایی اطلاق می‌شود که از مقادیر تصادفی برای حل مسائل استفاده می‌کنند.

هدف ← افزایش کارایی
 ← کاهش کمالات ← مثلاً در رمزنگاری

نکته: الگوریتم‌های تصادفی بر اساس میزان دقت در خروجی، دو رده کلی دارند:

* موارد جواب صحیح برمی‌گردانند.

* باید احتمال مشخص جواب درست می‌دهند و ممکن است جواب اشتباه نیز تولید کنند.

در این مثال می‌بینیم که تصادفی ساز در الگوریتم سبب افزایش کارایی می‌شود.

⑬ مثال برای الگوریتم‌های تصادفی:

توان مسائل: مرتب‌سازی لیست از اعداد

الگوریتم حل مسائل ← مرتب‌سازی سریع (Quicksort)

بهترین حالت $\frac{n^2}{2}$
 حالت میانه‌ای $n \log n$ ← بهینه‌ترین
 بهترین حالت $n \log n$
 بهینه‌ترین حالت ← Randomized Quicksort

