

⑨ جمع نپذیر - گام های حل مسائل

- ① بررسی دقیق مسأله و در نظر گرفتن حالت های خاص، فرضیات و ...
- ② پیدا کردن یک راه حل برای مسأله (گام نوآوری)
- ③ بیان رسمی (formal) راه حل مسأله (الگوریتم)
- ④ نشان دادن صحت و درستی الگوریتم ارائه شده
- ⑤ تحلیل کارایی الگوریتم
- ⑥ پیاده سازی الگوریتم

در درین طراحی
الگوریتم بیشتر
پرواضحتی است

← محوریت درس

⑩ انواع الگوریتم ها از حیث رویکرد حل مسأله :

- ① الگوریتم های قطعی (Deterministic Algorithms)
- ② الگوریتم های تقریبی (Approximation Algorithms)
- ③ الگوریتم های تصادفی (Randomized Algorithms)

⑪ الگوریتم های قطعی : به الگوریتم های اطلاق می شود که اگر در یک وضعیت مشخص باسیم، آنگاه با توجه به شرایط قطعی، وضعیت بعدی کاملاً مشخص است.

در مقابل الگوریتم های قطعی ← الگوریتم های غیر قطعی : به الگوریتم های اطلاق می شود که اگر در یک وضعیت مشخص باسیم، آنگاه با توجه به شرایط قطعی، چندین وضعیت بعدی وجود دارد و می توان به طور همزمان به همه وضعیت های بعدی انتقال یافت.

Non-deterministic Algorithms

چندین امکانی تنها در تئوری مطرح است.
تأیید

(۱۲) مثال برای الگوریتم‌های قطعی و غیر قطعی:

عنوان مسأله: جستجو $A[1..n]$
صورت مسأله: یک آرایه از اعداد (نه لزوماً مرتب) داده شده است، قرار است یک مقدار خاص مثل x را در آن جستجو کنیم.

Dsearch ($A[1..n], x$)

► Array A is not necessarily sorted.

1. for $i = 1$ to n do

2. if $A[i] == x$ then

3. return TRUE;

4. return FALSE;

الگوریتم قطعی

$O(n)$ مقیاس

n تا مقایسه در بدنه حلقه

برای بیان الگوریتم غیر قطعی، نیازمند معرفی یک سری $\left\{ \begin{matrix} \text{تابع} \\ \text{(دستور)} \\ \text{قرار داد} \end{matrix} \right\}$ جدید هستیم.

* تابع $choose(i)$: این تابع به طور همزمان i آگهی از سیستم ایجاد کرده و هر یک را به یکی از عناصر مجموعه S که اختصاص می‌دهد.

* تابع $success()$: این تابع به معنای توقف الگوریتم به همراه پیدا کردن جواب است.

* تابع $failure()$: این تابع به معنای توقف الگوریتم به همراه اعلام عدم پیدا کردن جواب است.

NDSearch ($A[1..n], x$)

► Array A is not necessarily sorted.

1. $i \leftarrow choose(S = \{1, \dots, n\})$

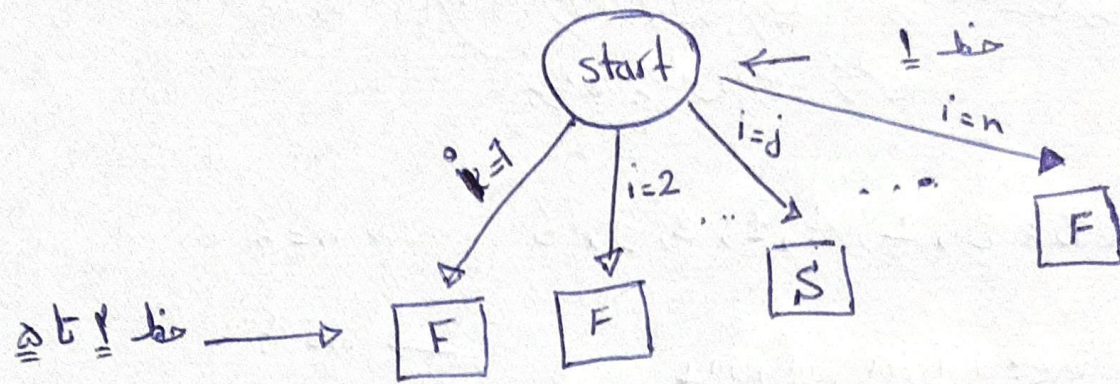
2. if $A[i] == x$ then

3. success()

4. else

5. failure()

۱۳) با فرض اینکه $A[n] = \infty$ است، نحوه اجرای الگوریتم NDsearch به صورت زیر است:



۱۴) الگوریتم‌های تقریبی (Approximation algorithms):

اقدام به حل مسأله به منظور دستیابی به یک جواب نزدیک جواب اصلی و نه دقیقاً جواب اصلی — مثلاً علت می‌تواند این باشد که حل مسأله برابر جواب دقیق کار است اما برای تکرار به طور کارایی جواب تقریبی بدست آورده.

۱۵) مثال برای الگوریتم‌های تقریبی:

عنوان مسأله: تخصیص پردازش

صورت مسأله: تقسیم n پردازش روی m ماشین به نحوی که زمان هر پردازش هدف آنجا کمینه شود.

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

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

m_1 | 3 | 3

m_2 | 2 | 3

m_3 | 4 | 6

m_4 | 1

راهنمای ۱: پردازش‌ها را به ترتیب بین ماشین‌ها

تقسیم کنیم