

# طراحی و تحلیل الگوریتم ها

مدرس: سعدون عزیزی

[s.azizi@uok.ac.ir](mailto:s.azizi@uok.ac.ir)

گروه مهندسی کامپیوتر

نیم سال دوم ۹۷-۹۶

# تکنیک‌های طراحی الگوریتم‌ها

- تقسیم و حل
- برنامه‌ریزی پویا
- الگوریتم‌های حریصانه
- الگوریتم‌های عقبگرد
- الگوریتم‌های شاخه و حد

# الگوریتم‌های حریصانه (Greedy Algorithms)

- الگوریتم‌های مسئله‌های بهینه‌سازی معمولاً مراحل مختلفی را طی می‌کنند و در هر مرحله مجموعه‌ای از انتخاب‌ها را پیش رو دارند.
- الگوریتم‌های حریصانه همیشه گزینه‌ای را انتخاب می‌کنند که در لحظه به نظر بهترین می‌آید.
- این نوع الگوریتم‌ها یک انتخاب بهینه محلی انجام می‌دهند به این امید که این انتخاب به یک جواب بهینه سراسری ختم شود.
- این نوع الگوریتم‌ها فرصت این را ندارند که به عقب برگردند و انتخاب خود را تغییر دهند
- الگوریتم‌های حریصانه همیشه به جواب بهینه ختم نمی‌شوند.

## مسئله انتخاب فعالیت

- ❑ فرض کنید یک مجموعه  $S = \{a_1, a_2, \dots, a_n\}$  از  $n$  فعالیت داریم که همه باید از یک منبع مشترک، مانند یک سالن سخنرانی، استفاده کنند.
- ❑ در هر زمان فقط یکی از فعالیت‌ها می‌تواند از منبع استفاده کند.
- ❑ هر فعالیت  $a_i$  دارای یک زمان شروع  $s_i$  و یک زمان پایان  $f_i$  است که در آن  $0 \leq s_i < f_i < \infty$ . فعالیت  $a_i$  در صورت انتخاب، در بازه نیمه باز  $[s_i, f_i)$  اجرا می‌شود.
- ❑ فعالیت‌های  $a_i$  و  $a_j$  سازگار هستند اگر بازه‌های  $[s_i, f_i)$  و  $[s_j, f_j)$  با یکدیگر تلاقی نداشته باشند (به عبارت دیگر،  $s_j \geq f_i$  یا  $s_i \geq f_j$ ).
- ❖ **مسئله انتخاب فعالیت** این است که یک مجموعه از فعالیت‌های دوبه‌دو سازگار با اندازه بیشینه انتخاب کنیم.

## مسئله انتخاب فعالیت

مثال: □

$i$	1	2	3	4	5	6	7	8	9	10	11
$s_i$	1	3	0	5	3	5	6	8	8	2	12
$f_i$	4	5	6	7	8	9	10	11	12	13	14

چند مجموعه از فعالیت‌های دوبه‌دو سازگار: □

- $\{a_3, a_9, a_{11}\}$
- $\{a_1, a_4, a_8, a_{11}\}$
- $\{a_2, a_4, a_9, a_{11}\}$

## مسئله انتخاب فعالیت: برنامه ریزی پویا

- واضح است که مسئله انتخاب فعالیت از زیرساختار بهینه پیروی می کند.
- فرض کنید  $S_{ij}$  نشان دهنده ی مجموعه فعالیت هایی باشد که بعد از پایان فعالیت  $a_i$  آغاز می شوند و قبل از فعالیت  $a_j$  پایان می یابند.
- حال می خواهیم یک مجموعه بیشینه از فعالیت های دوبه دو سازگار در  $S_{ij}$  بیابیم. فرض کنید چنین مجموعه ی بیشینه ای شامل فعالیت  $a_k$  باشد.
- بنابراین، دو زیرمسئله برای ما باقی می ماند: یافتن فعالیت های دوبه دو سازگار در مجموعه  $S_{kj}$  و  $S_{ik}$ .
- فرض کنید که  $c[i,j]$  اندازه یک جواب بهینه برای مجموعه  $S_{ij}$  باشد.  $c[i,j]$  از رابطه بازگشتی زیر قابل محاسبه است:

$$c[i, j] = \begin{cases} 0 & \text{if } S_{ij} = \emptyset, \\ \max_{a_k \in S_{ij}} \{c[i, k] + c[k, j] + 1\} & \text{if } S_{ij} \neq \emptyset. \end{cases}$$

## مسئله انتخاب فعالیت: الگوریتم حریصانه

- منظور از انتخاب حریصانه برای مسئله انتخاب فعالیت چیست؟
- فعالیتی را از مجموعه  $S$  انتخاب کنیم که بیشترین منبع ممکن را برای فعالیت‌های دیگر باقی می‌گذارد. به عبارت دیگر، فعالیتی که زودترین زمان پایان را دارد.
- اگر بیش از یک فعالیت در  $S$ ، زودترین زمان پایان را داشت، آنگاه می‌توان به دلخواه یکی را انتخاب کرد.
- برای این منظور، فعالیت‌ها را به ترتیب صعودی زمان پایان مرتب می‌کنیم.
- اگر انتخاب حریصانه را انجام دهیم، در هر مرحله فقط یک زیرمسئله برای حل کردن باقی می‌ماند.
- **یک سوال اساسی:** آیا این رویکرد ما را به جواب بهینه می‌رساند؟

## مسئله انتخاب فعالیت: الگوریتم حریصانه

**قضیه:** یک زیرمسئله ناتهی  $S_k$  را در نظر بگیرید و فرض کنید  $a_m$  یک فعالیت در  $S_k$  باشد که کمترین زمان پایان را دارد. در این صورت، یک زیرمجموعه بیشینه از فعالیت‌های دوبه‌دو سازگار در  $S_k$  وجود دارد که شامل  $a_m$  است.

**اثبات:** فرض کنید  $A_k$  یک مجموعه بیشینه از فعالیت‌های دوبه‌دو سازگار در  $S_k$  باشد و فرض کنید که  $a_j$  فعالیتی در  $A_k$  با زودترین زمان پایان باشد.

□ اگر  $a_j = a_m$  باشد، کار ما تمام است.

□ اگر  $a_j \neq a_m$  باشد، مجموعه  $A'_k = A_k - \{a_j\} \cup \{a_m\}$  را می‌سازیم. بدیهی است که فعالیت‌های درون  $A'_k$  سازگار هستند (چون  $f_m \leq f_j$ ). توجه داشته باشید که  $A'_k = A_k$ . نتیجه می‌گیریم که  $A'_k$  یک زیرمجموعه بیشینه از فعالیت‌های دوبه‌دو سازگار در  $S_k$  است، شامل فعالیت  $a_m$ .



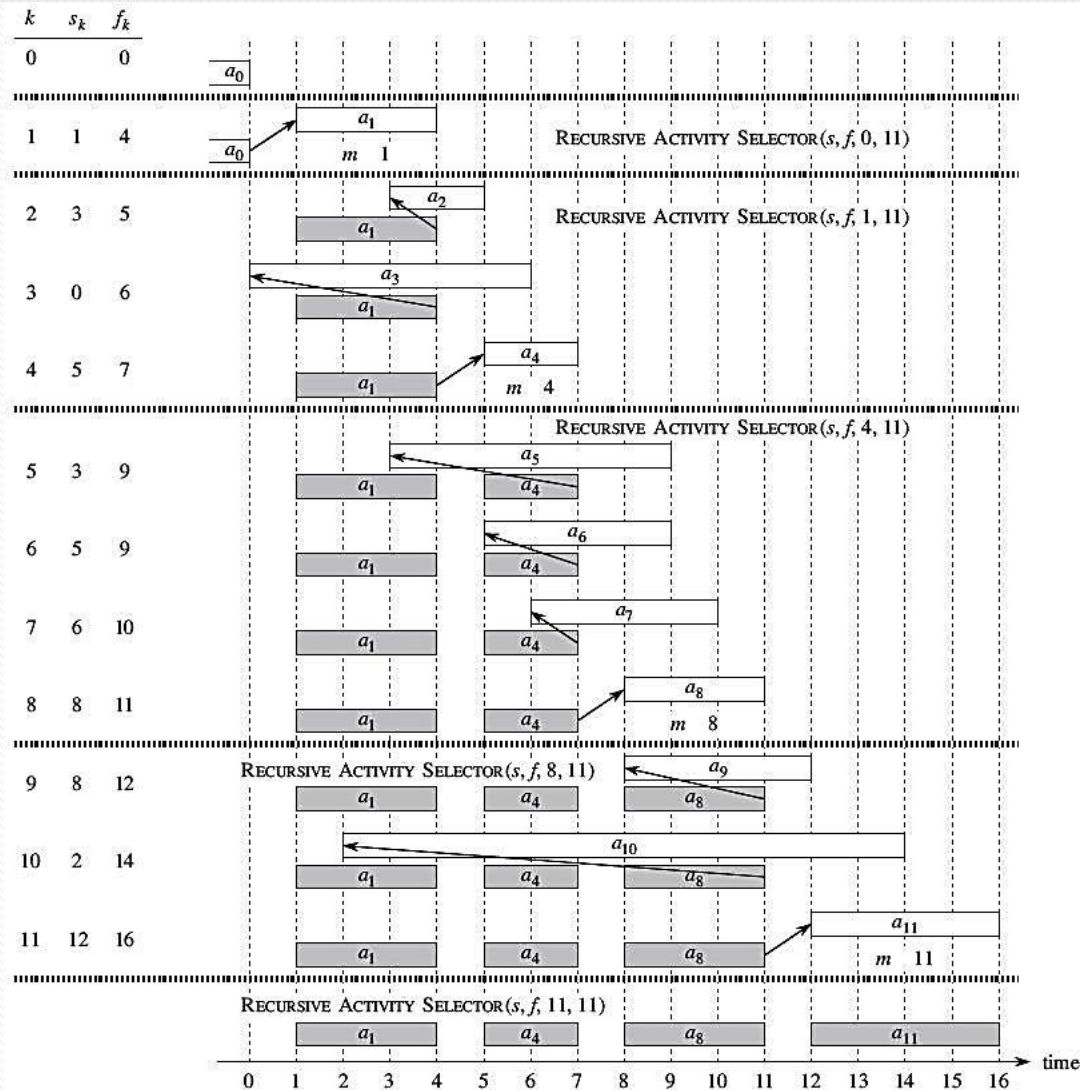
## مسئله انتخاب فعالیت: الگوریتم بازگشتی

RECURSIVE-ACTIVITY-SELECTOR( $s, f, k, n$ )

```
1   $m = k + 1$ 
2  while  $m \leq n$  and  $s[m] < f[k]$            // find the first activity in  $S_k$  to finish
3       $m = m + 1$ 
4  if  $m \leq n$ 
5      return  $\{a_m\} \cup \text{RECURSIVE-ACTIVITY-SELECTOR}(s, f, m, n)$ 
6  else return  $\emptyset$ 
```

RECURSIVE-ACTIVITY-SELECTOR( $s, f, 0, n$ ) فرخوانی اولیه به صورت است.

# مسئله انتخاب فعالیت: الگوریتم بازگشتی



## مسئله انتخاب فعالیت: الگوریتم تکراری

GREEDY-ACTIVITY-SELECTOR( $s, f$ )

```
1   $n = s.length$ 
2   $A = \{a_1\}$ 
3   $k = 1$ 
4  for  $m = 2$  to  $n$ 
5      if  $s[m] \geq f[k]$ 
6           $A = A \cup \{a_m\}$ 
7           $k = m$ 
8  return  $A$ 
```