چند تست کنکور 98

۷۰- مرتبه زمانی الگوریتم زیر کدام است؟

```
f(n)
{
    x = ۲; y = ۱;
    while(y <= n){
        y = y × x;
        x = x × x;
    }
}
```

۱) $n\log n$ ۲) $\log n \log n$ ۳) $\log(\log n)$ ۴) $\log(\log(\log n))$

۷۲- فرض کنید زیر برنامه $C = aux(A,B)$ دو ماتریس A و B با اندازه $n \times n$ را ضرب کرده و نتیجه را در C برمی‌گرداند. مقدار برگشتی تابع زیر برای ماتریس M کدام است؟

```
Mat(M)
{
  P = M ; Q = M
  for i = ۱  to n − ۱ do{
      P = aux(P,M)
      Q = aux(Q,P)
  }
  return(Q)
}
```

۱) M^n ۲) M^{n-1} ۳) $M^{\frac{n(n-1)}{2}}$ ۴) $M^{\frac{n(n+1)}{2}}$

۷۹- اگر $T(n) = 2T(\frac{n}{4}) + \theta(1)$ از مرتبه $\theta(\sqrt{n})$ باشد، آنگاه $T(n) = 2T(\frac{n}{4}) + \theta(\sqrt{n})$ از کدام مرتبه است؟

۱) $O(n)$ ۲) $O(\log^2 n)$ ۳) $O(n\sqrt{\log n})$ ۴) $\theta(\sqrt{n}\log n)$

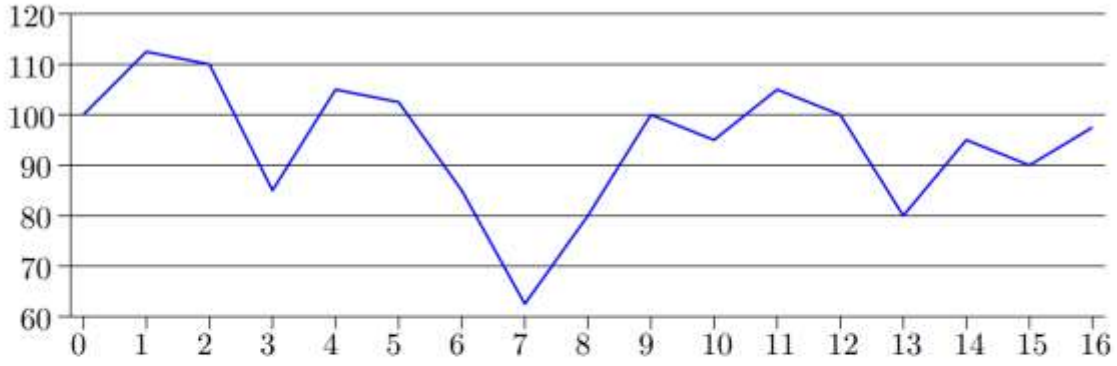
۸۰- اگر $U(m) = T(2^m)$ باشد و $T(n) = T(\frac{n}{2}) + \log n$، آنگاه رابطه بازگشتی U کدام است؟

۱) $U(m) = U(m-1) + 1$ ۲) $U(m) = U(m-1) + m$
۳) $U(m) = U(m-1) + \log m$ ۴) $U(m) = U(m-1) + 2^m$

Maximum Subarray Problem

Example: I am giving you perfect stock market predictions of a company ABC for the next thirty days. But under a condition — you can buy and sell the stocks only once. That's it. You can't do more than one "buy-sell" transaction. What would be the best day to buy the stocks? And when do you need to sell them to make the maximum profit?

Buying and Selling



day	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
price	100	113	110	85	105	102	86	63	81	101	94	106	101	79	94	90	97
change		13	-3	-25	20	-3	-16	-23	18	20	-7	12	-5	-22	15	-4	7

Problem

When were the best times to buy and sell?

- A Brute-force Solution
  - Try every possible pair of buy and sell dates (buy precedes sell date)
  - Period of n days has $\binom{n}{2}$ pairs of dates $\binom{n}{2}$ is $\Theta(n^2)$
  - Best we could hope is to evaluate each pair in constant time and approach would take $\Omega(n^2)$
  - Can we do better?

- Difficulty: We don't necessarily buy at the lowest price or sell at the highest price.
- Example:

Problem Transformation

- Consider the array $A[1 \ldots 16]$ of change in price.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
13	-3	-25	20	-3	-16	-23	18	20	-7	12	-5	-22	15	-4	7

Maximum subarray $A[8 \ldots 11]$

- The *maximum subarray* is the contiguous subarray whose elements have the largest sum. Here, it is $A[8 \ldots 11]$.
- So the best times to buy and sell are days 7 and 11.
- We reduced our original problem to:

Problem

Given an array $A[1 \ldots n]$ of n numbers, find the indices p, q such that $1 \leqslant p \leqslant q \leqslant n$ and $\sum_{i=p}^{q} A[i]$ is maximum.

The Maximum Subarray Problem

Brute-force approach to MAXIMUM SUBARRAY

```
1:  procedure MAXIMUM-SUBARRAY(A, 1, n)
2:      max ← −∞
3:      for i ← 1, n do
4:          for j ← i, n do
5:              sum ← 0
6:              for k ← i, j do
7:                  sum ← sum + A[k]
8:              if sum > max then
9:                  max ← sum, p ← i, q ← j
10:     return p, q, max
```

$\sum_{i=1}^{n} \sum_{j=i}^{n} \sum_{k=i}^{j} (1) = O(n^3)$

Kadane $O(n)$

- Running time? $\Theta(n^3)$.

Applications

The maximum subarray problem has several applications. Some of the well-known applications are in genomic sequence analysis and computer vision. They are used in genomic sequence analysis to identify important segments of protein sequences like GC-rich regions, and regions of high charge. In computer vision, they find their use in detecting the brightest area in bitmap images.

مثال ۱-۱۹ الگوریتم زیر برای تعیین عنصر بیشینهٔ آرایهٔ $A[1..n]$ مفروض است. این الگوریتم، اندیس عنصر بیشینه را توسط j برمی‌گرداند. با فرض آن‌که عناصر آرایهٔ A اعداد صحیح متمایز و متساوی‌الاحتمال هستند، میانگین زمان اجرای دستورالعمل d یعنی $j = i$ را به‌دست آورید.

```
Algorithm FindMax ( A, n, j )
a)      j = n
b)      for i = n-1 downto 1 do
c)          if A[i] > A[j] then
d)              j = i
```

حل. تعداد دفعات اجرای دستورالعمل‌های الگوریتم فوق در بدترین حالت، بهترین حالت و حالت میانگین، در زیر آمده است.

دستورالعمل	تعداد دفعات اجرا در بدترین حالت	تعداد دفعات اجرا در حالت میانگین	تعداد دفعات اجرا در بهترین حالت
a	۱	۱	۱
b	$n-1$	$n-1$	$n-1$
c	$n-1$	$n-1$	$n-1$
d	$n-1$	$A_n = ?$	۰

با توجه به جدول مزبور مشاهده می‌شود که a_n یعنی میانگین زمان اجرای دستورالعمل $j = i$ در رابطهٔ $0 \le A_n \le n-1$ صادق است. برای روشن شدن مطلب به بررسی حالت خاص $n = 3$ می‌پردازیم. با توجه به متمایز بودن عناصر آرایهٔ A، حالت‌های امکان‌پذیر برای آرایش عناصر آن به‌صورت‌های زیر هستند.

U_n : میانگین دفعات اجرای دستور در آرایه n تایی

$$U_n = \sum_{i=1}^{n} (U_{i-1}+1)\frac{1}{n}$$

$$n U_n = \sum_{i=1}^{n} (U_{i-1}+1) \Rightarrow n U_n - (n-1)U_{n-1} = U_{n-1}+1$$

$$(n-1)U_{n-1} = \sum_{i=1}^{n-1} (U_{i-1}+1)$$

$$n U_n - n U_{n-1} = 1$$

$$U_n = U_{n-1} + \frac{1}{n}$$

$$U_n = \ln \frac{1}{2} + \cdots + \frac{1}{n} = O(\log n)$$