به نام پروردگار زیبایی

محمد عرفان زارع زردینی

Subject ۹۸۴۲۴۴۳۲

Date

کوئیز اسلاید 5

(Q1

۸ و ۱۵ و ۱۰ و ۲۰ و ۱

Sodacode:

```
for i ← 1 to n-1
    Smallest ← j
    for j ← i to n
        if A[j] < A[Smallest]
            Smallest ← j
    Exchange A[i] ↔ A[Smallest]
```

کوچکترین
مرحله ۱: ۸ و ۱۵ و ۱۰ و ۲۰ و ۱۰ — ۱۰ با ۵ جابجا شد

دومین کوچکترین مرحله ۲: ۸ و ۱۵ و ۲۰ و ۱۵ و ۱۰ — ۱۵ با ۸ جابجا شد

⇐

سومین کوچکترین مرحله ۳: ۲۰ و ۱۸ و ۱۵ و ۱۰ — ۱۸ با ۱۵ جابجا شد

گزینه A درسته

به روش دوم با انتخاب max: مرحله ۱: ۲۵ و ۱۰ و ۱۸ و ۱۵ — ۲۰ با ۲۵ جابجا شد؛ مرحله ۲: ۲۵ و ۱۸ و ۱۰ و ۱۵ — ۱۸ با ۱۵ جابجا شد

مرحله ۳: ۲۵ و ۱۸ و ۱۵ و ۱۰ و ۵ — ۱۵ با ۵ جابجا شد

(Q2

در Selection Sort تعداد دورهای کامل یک (تعداد Passes) به for اصلی وابسته دارد و در نتیجه با توجه به اینکه

حلقه تعداد بار انجام for اصلی n-1 می‌باشد پس در اینجا جواب ۱۹=۱-۲۰ می‌باشد پس B درسته

(Q3 در هر صورت با توجه به Sodacode آن که در سوال 1 نوشته شده، در for داخلی مقایسه‌ها

انجام می‌شود (حذف E). i از ۱ تا ۶ می‌رود. در for داخلی هم j از i تا ۷ می‌رود. در نتیجه داریم: همچنین با استفاده از

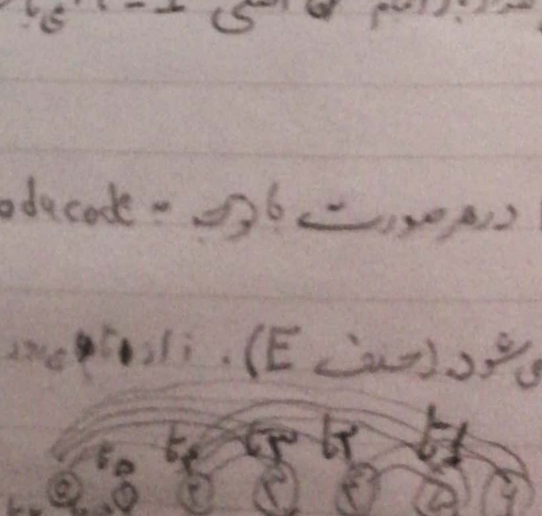
$i=1 \to 6$

$6+5+4+3+2+1=21$

$\sum_{i=1}^{6} i = \frac{(1+6)\times 6}{2} = 21$

فرمول گاوس

شکل داریم:



Q۴) با توجه به اینکه در صورت استفاده از selection sort به صورت Largest in last باشد،

در نتیجه ما پس از ۴ مرحله باید ۴ عنصر max در بین اعداد در جای اصلی خود باشند که اینگونه نیست پس

از این روش (استفاده شده (حذف B)) پیرامون insertionSort اما، مشکلی وجود ندارد

و چون این روش، اندیس ها اندیس جلو می رود و مرتب می کند، گویی ۴ عنصر اولیه موجود را مرتب کرده و پس خود

سپس سراغ عناصر باقی مانده می رود و آن ها را نیز مرتب و در جای خود قرار می دهد و در نتیجه، ترتیب

تولید شده تا الان (پس از ۴ مرحله) تناقضی ندارد و او کیه می تواند insertion sort باشد.

در نتیجه گزینه ۲ درست است.