بسم الله الرحمن الرحیم

Subject: ۹۸۴۱۱۴۳۲

محمد عرفان زارع دربیدی

①

a) worst case
- insertion sort: $O(n^2)$
- Heap Sort: $O(n \log(n))$
- Prims: $O(|E| \log|E|)$

b) $f(n) = n^2 \log n^2 + 2n \lg^2 n$

$n^2 > 2n \quad n^2 \log n^2 > 2n \log n^2 \Rightarrow 2n^2 \log n > 2n \log n \Rightarrow n^2 \log n^2 > 2n \log n$

$n > \log n$

یعنی در مقدار بالا اردر ($n^2 \log n^2$) است.

c) $f(n) = (n(100n + 5 + n^3))^2$

$= (100n^2 + 5n + n^4)^2 = n^8 + 200n^6 + 10n^5 + 10000n^4 + 1000n^3 + 25n^2$

⇐ اردر صعود و افزایش حجم بالای داده‌ها را داریم در n^8 است پس $O(n^8)$

d) $f(n) = f(\frac{n}{3}) + f(\frac{n}{4}) + \theta(n^{\sqrt{\log n}})$

از $f(\frac{n}{3}) \Rightarrow a=1, b=3 \quad n^{\log_b^a} = n^0 = 1 \qquad 1 < n^{\sqrt{\log n}}$

$\varepsilon = \sqrt{\log n} > 0$

$a f(\frac{n}{b}) \le c f(n) \Rightarrow 1 (\frac{n}{3})^{\sqrt{\log \frac{n}{3}}} \le c n^{\sqrt{\log n}} \Rightarrow C < 1 \Rightarrow n^{\sqrt{\log n}}$ اردر

$\log \frac{n}{3} = \log n - \log 3$

از $f(\frac{n}{4}) \Rightarrow a=1, b=4 \quad n^0 = 1 \qquad 1 < n^{\sqrt{\log n}} \quad \sqrt{\log n} = \varepsilon > 0$

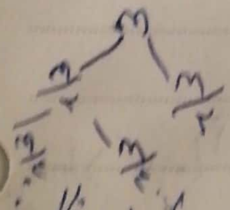
$\Rightarrow 1 (\frac{n}{4})^{\sqrt{\log \frac{n}{4}}} \le c n^{\sqrt{\log n}} \Rightarrow c < 1 \Rightarrow n^{\sqrt{\log n}}$

PAPCO

۲) collision رخ ندهد

الف) غلط است. چون باید بخش های قبل پر شده باشند تا ... و از نوع search است عملیات مقدار مورد نظر انتظار زمان از اردر $O(n/m)$ است. ابتدا باید پر شوند بعد collision رخ بده

ب) بله. چون هر بار دارند به دو قسمت تقسیم می‌شوند نصف نصف می‌شود

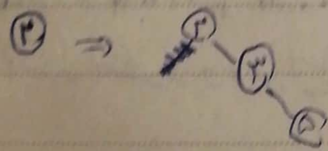


درنتیجه درسته.

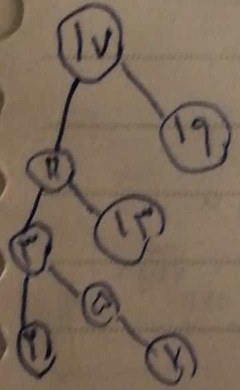
ت) غلط. درست است که HeapSort in place عمل می‌کند ولی یکی از نکات و ویژگی های Radix sort که stable sort بودنش است که در مرتب سازی مهم است ولی در HeapSort این ویژگی در نظر گرفته نشده و رعایت نمی‌شود

۳)

الف)



از این که فقط یک عدد بزرگتر از Root داریم می‌فهمیم که Root یکی مونده به آخری است و ۱۷



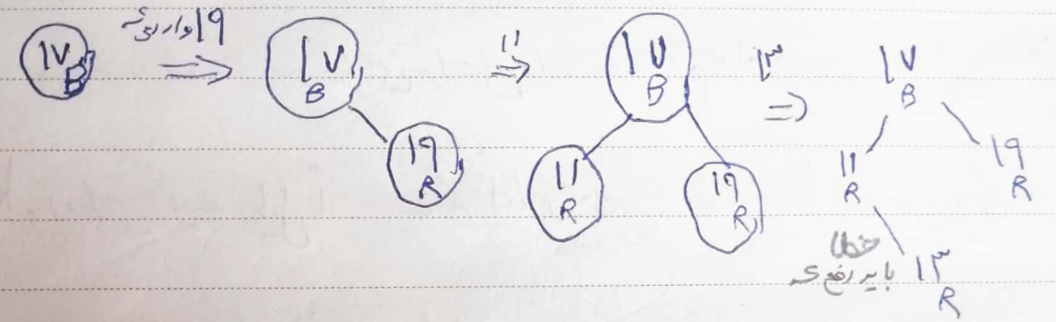
و از آنجا که فرزند چپ Root یک عدد بزرگتر دارد می‌فهمیم ۱۱ است و عدد دیگر ۱۳ است.
همین را برای بقیه هم می‌رویم و شکل روبرو حاصل می‌شه

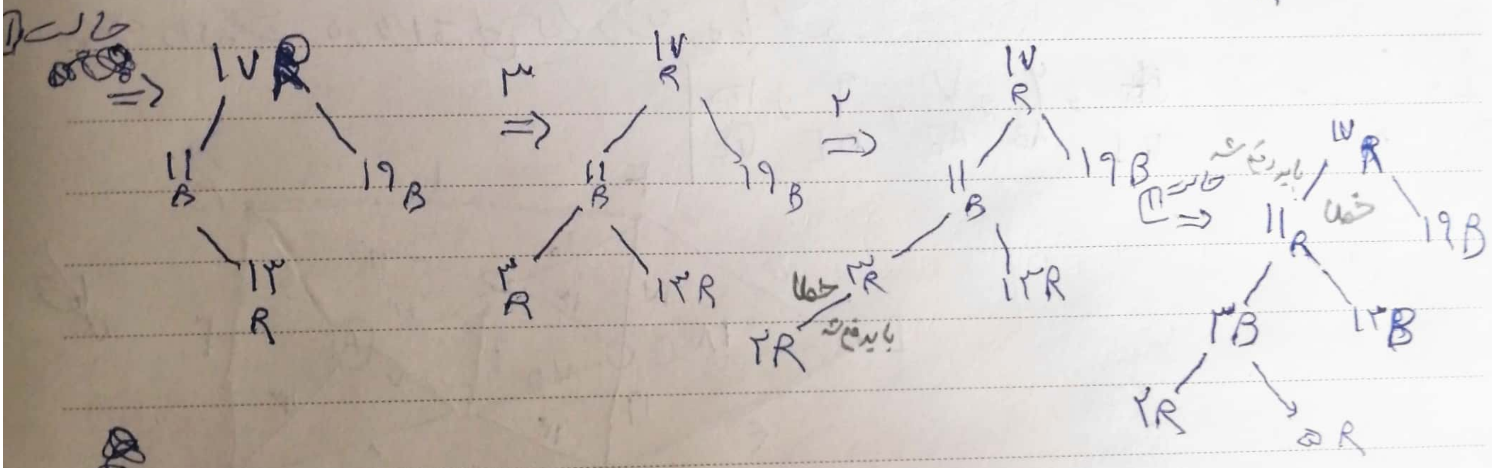
۳) ب)

Root باید سیاه باشه

فرض میکنیم ترتیب واردشدن به درخت براساس بخش الف و رسم نمایی باشد

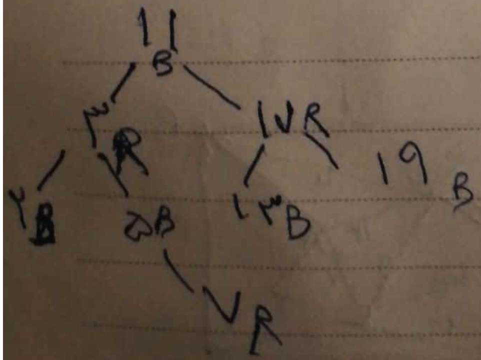
⇒ ۱۷ و ۱۹ و ۱۱ و ۱۳ و ۳ و ۲ و ۵ و ۷





علت این که در ختان R-B Tree میشه ایجاد چنین خطایی است که باید
با چرخش و حل شود

ج) چرخش بین ۱۷ و ۱۱



Subject:
Date

ص ۴

۴)

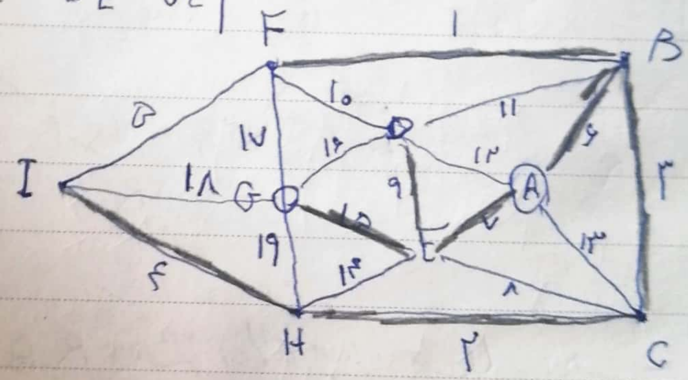
الف)

به علت اینکه باید شکل رو شلوغ نکنم تمامی مراحل را توضیح دادم و از روی خود داده‌ها رفتم - البته ترتیب را اعمال کنم

روش kruskal بر اساس کمینه یال است که ایجاد cycle نکند و تا آنجایی بریم که همه رئوس پوشش داده شده و حلقه‌ای ایجاد نشده باشد.

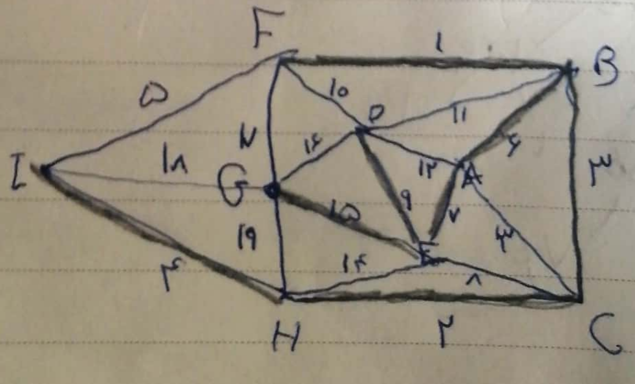
| 1 , 2 , 3 , 4 , ۶ , ۷ , ۹ , ۱۵ |
FB, HC, BC, HI, AB, AE, DE, GE

شکل



ب) از روی شکل و قدم به قدم جلو می‌رویم

۱۵ و ۹ و ۷ و ۴ و ۲ و ۳ و ۱ و ۶



PAPCO

ص ۵

Subject:
Date

۵)
وقت نشد

۶)

آرایه G و S را مرتب صعودی میکنیم ✓

G(n) به معنای عنصر n ام آرایه G است =

```
size = Length(S)
bachShi = [0, 0]
while(size > 0)
    while (G(bach) > 0)
        { G(bach)-- = S(Shirin)
          size--;
          Shirini + 4
        }
    if (G(bach) <= 0) bach = 1
    Return bach + 1
```

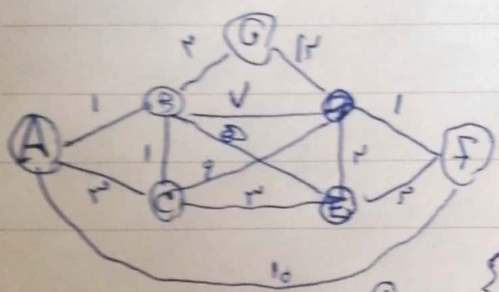
با شروع از بچه های کوچکتر به جواب بهتری میرسیم چون تا حد ممکن در مصرف شیرینی صرفه جویی میشود.
و در نتیجه با شیرینی کمتری بچه ها راضی میشوند

ب) در حلقه فقط با size کار داریم پس اُردر o(n) است.
البته اگر با مرتب سازی باشد که سورت O(nlogn) است

PAPCO

ج) این الگوریتم تا حدی بهینه است چون از کوچکترین به
بزرگترین می رویم و با حداقل پیمایش به بزرگترین می رسیم

۷)



A { AB = ۱ ✓ (۱) ; AC = ۳ (۲) ; AF = ۱۰ }

B ⇒ { BG = ۳ (۳) ⇒ G { GD = ۱۵ } ; BD = ۸ (۴) ⇒ D { DF = ۹ > ۸ ✗ ; DE = ۱۰ > ۸ ✗ } ; BC = ۲ < ۳ ⇒ ok }

{ CE = ۵ > ۸ → ok ; CD = ۱۱ > ۸ ✗ }

C = { CD = ۱۲ > ۸ ✗ ; CB = ۴ > ۱ ✗ ; CE = ۴ (۵) ✗ }

E { ED = ۷ ; EF = ۸ < ... ⇒ ok }

با توجه به اینکه تا وزن ۹ پیش رفتیم و کمترین هزینه هم ۸ بوده دیگر نیازی به جلو رفتن نیست
و مسیر پیدا شده.

کم‌هزینه‌ترین مسیر از A تا F : ABCEF = ۸ البته ACEF هم و...

رأس شناخته شده: A, B, C, E, D, G, F

۸) بن بست

A — B (۱) ۲ — D — F (۲) ۴ — G (۳) ۸

A — C (۴) ۷ — D ۶ > ۴ ✗

A — E ۱۲

E — ۹ < ۱۲ ok — G → ۲ < ۸ ok — D → (۴) ۴ < ۶ ok → F ۵ < ۶ ok

B ۱۰ > ۱ ✗ ; A ۵ > ۰ ✗

* درصورتی که راه‌گذاری‌ها ترتیب جلو رفتن است
در این نمی‌شود جواب بهینه را بدست آوریم

G ↓ ۷ < ۸ ok

بن بست
تمام مقادیر و مراحل نوشته شد

Subject:

Date

ادامه ۸

ب) برای F و G ⇐ ۳ = ها (یال منفی) ... ABD ۴ : D برای

همه به علت وصل بودن به D رخ داد

ج) یال های منفی را الگوریتم جواب درست می دهد.

E G, ·E A, CD

چون با الگوریتم همخوانی ندارند.