(جلسه 2)

- 2 مسئله حائز اهمیت است. میزان لودی که در حین اجرا می‌شه و میزان لودی که در حافظه انجام می‌شه.

- حلقه چند بار تکرار می‌شه

- بهترین حالت، بدترین حالت، همون امکانات تکرار

$ax^2+bx+c=0 \Longrightarrow$ از مرتبه 2

```
n بار ← for (i=0, i<=n, i++)
m بار ← for (j=0, j<=m, j++)
            sum += a[i][j];
```

if $n \simeq m \Rightarrow$ ~~n^2~~ بار

- یک الگوریتم بهینه باشه از لحاظ تعداد اجرا مقایسه

- برای ۷ بهینگی ~~[illegible]~~ یک تابع ایجاد کنم و آن دو تا تابع را با هم مقایسه کنم

- پیچیدگی ~~[illegible]~~ ؟ پیچیده بودن تابع ریاضی را رابطه ی مستقیم دارد.

ما به دنبال راهکاری می‌گوییم که مسئله‌های بیابیم

بیشتر از مسئله‌های قبلی (حتی با کمی بیشتر بودن)

سپس این بحث سر BIG Data هست چون

در داده های بزرگ کمی بیشتر بودن الگوریتم زمان

بسیار بیشتری را خواهد داد. این بحث های که داریم

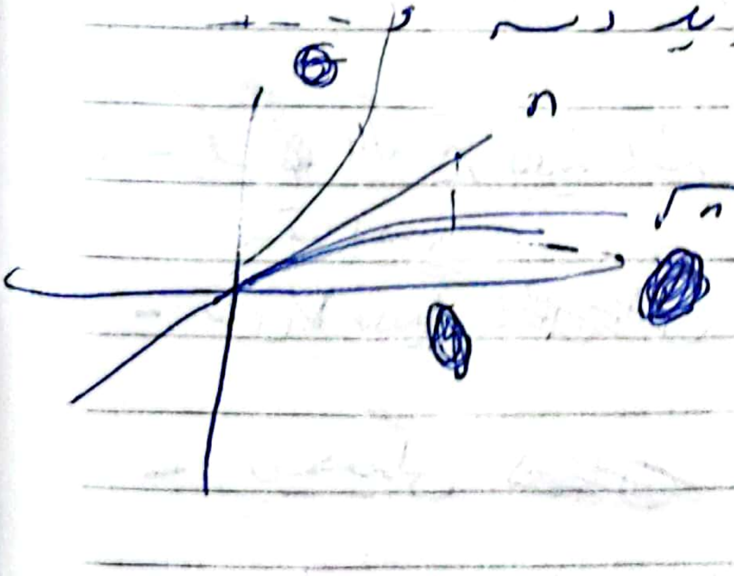
سر داده هست نه در کدها.

مثلاً در دیکشنری اینکه ما بیاییم و

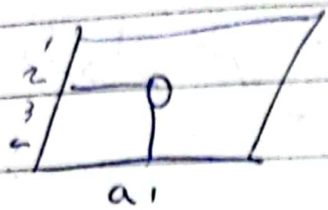
دنبال یک کلمه بگردیم کار را روی چینش

داده انجام می‌دیم که تمام a ها در یک

دسته b ها در یک دسته

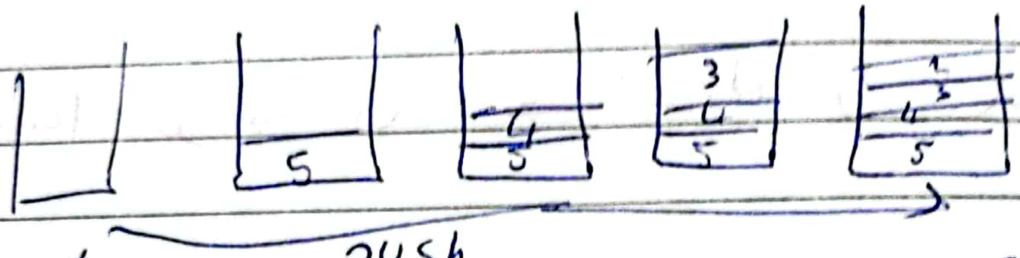


ما وزن برنامه را به حافظه و cpu باید بدیم که نسبت بهینه است
مثلاً برای فاکتوریل



مثلاً توی ۱۰۰! الان این روش خیلی بهتره تا سریعاً حل کنه.

مثل stack در



push

مرحله اول خالی

استک در memory است.

n بار push.

n خانه‌ی حافظه مصرف کردیم

در اصل هر متغیری فراخوانی می‌کنی یک push است.

فراخوانی تابعی با پارامتر n ؛

fact(a)

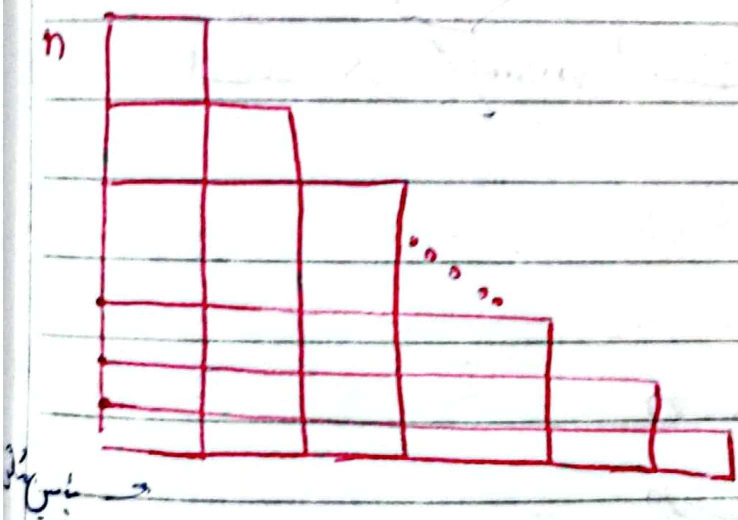
حالا با روش stack در محاسبه‌ی فاکتوریل:

$fact(a) = a \times fact(a-1)$

دیگر سرریز نمی‌شیم و بار را انداختیم روی روش memory

حالا اگر بار را روی دو تا CPU بخواهیم
بندازیم که مسئله ا
← تاریخ تحویل دوشنبه

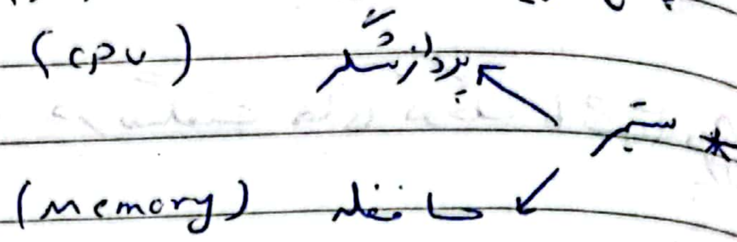
تمرین: در پایگاه داده ی روبه رو که مرتب سازی اجرا
در حالت بر مبنای load و CPU و
بر مبنای load حداقل را محاسبه کنید.



هر دو که باشد و محاسبات مرتبه
اجرا را بنویسید

Subject: ...... Date: / /

پاسخ تمرین توسط استاد (و مرور) :



کار پردازش را می‌توان روی دوش حافظه یا cpu انداخت.

تکنولوژی GOPU : برای مدیریت load balacing

اسم سرعت در cpu : Gig ← ۱۰۰۰ برابر سریع‌تر از ram
" " در memory : meg

می‌یاد جدول lacub درست می‌کنه و اونجا پردازش رو انجام می‌ده.
اشتباه

الگوریتم بهتر دست بالاتر داره [redacted] جهت‌ها.

a = b + c ← ترن دارن. اولی سه تا متغیر
یا
① a = a + c ← را خوانده و دوتا دوتا.
بهتره از اولی

کارت CD با usb : usb ، random acssess memmory دست

ولی CD ، سیکونشال است باید دور به دور بره

بنویسیم تا به هدف برسیم.

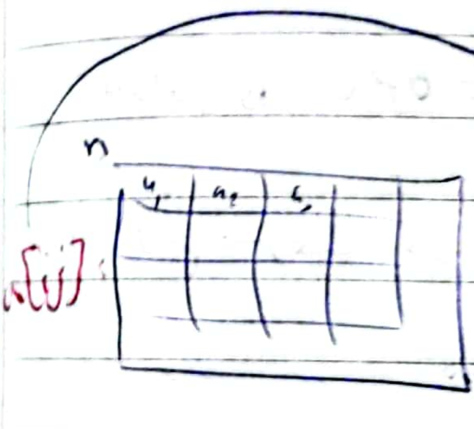
Niko

الگوریتم بهتر

۱- اولویت روی cpu باشه چون سریعتر است.

۲- اگر در حافظه هست در سگمنت های کوچک باشه تا چیزی کوتاه باشه.

پایین ترین ترتیب استاد:



آرایه مربعی : $n \times n$

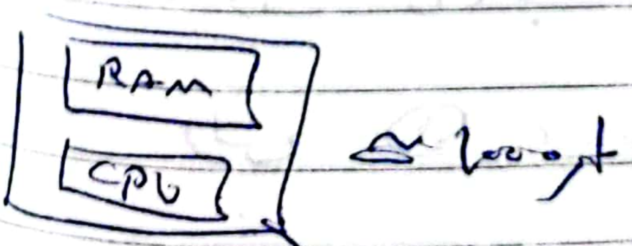
$a = b + c$

$a = a + c$

```
for i = 0 ; i < n ; i++ ;   → n بار
    for j = 0 ; j <= n ; j++ ;   → n بار
```

load در RAM

$n \times n = n^2$ → load روی cpu



اینکه مثل RAM روی ... تاثیر نذارد

Subject:

Date: / /

محاسبه ماتریس به ماتریس : این در GPU هست که ما نیاز داریم تصویر کار کنیم و فریم به فریم روی صفحه بیاریم.

$n^2 \times GHz \rightarrow CPU$

$n^2 \times MHz \rightarrow RAM$

پیکانی:

$$\frac{n \times (n-1)}{2}$$

load روی CPU GHz

load روی RAM MHz

Niko