

$$n! = \begin{cases} 1 & n=1 \\ (n-1)! & n>1 \end{cases}$$

$$fib \Rightarrow fib_n = fib_{n-1} + fib_{n-2}$$

$$fact(a) = a \times \underbrace{fact(a-1)}$$

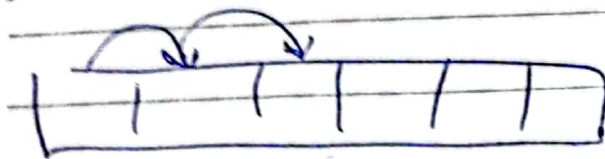
سوی P.B طول به اندازه چون هر کدام به مقدار  
بیشتر عایتیم می شه:

اولین کار در الگوریتم ساده کردن متغیرها  
است.

باید تمامی متغیرها <sup>فرد</sup> ~~زوج~~ تبدیل به ~~متغیر~~ <sup>جمع</sup> کنیم.

$$3 \times 100 = \underbrace{3 + 3 + \dots + 3}_{100 \text{ بار}}$$

سریع ترین : هر دو مرتبه با هم مقایسه شود



② merge sort : با اندیس گذاری کار دارد

Selection sort (انتخابی)  
 ایجاد max و توهم هر مرحله (مقایسه)  
 بدایه و به آخر تراز ده شده:  
 3 5 2 4  
 3 4 2 5  
 3 2 4 5  
 2 3 4 5

③ لیست پیوندی

همه ی 46 یادی است

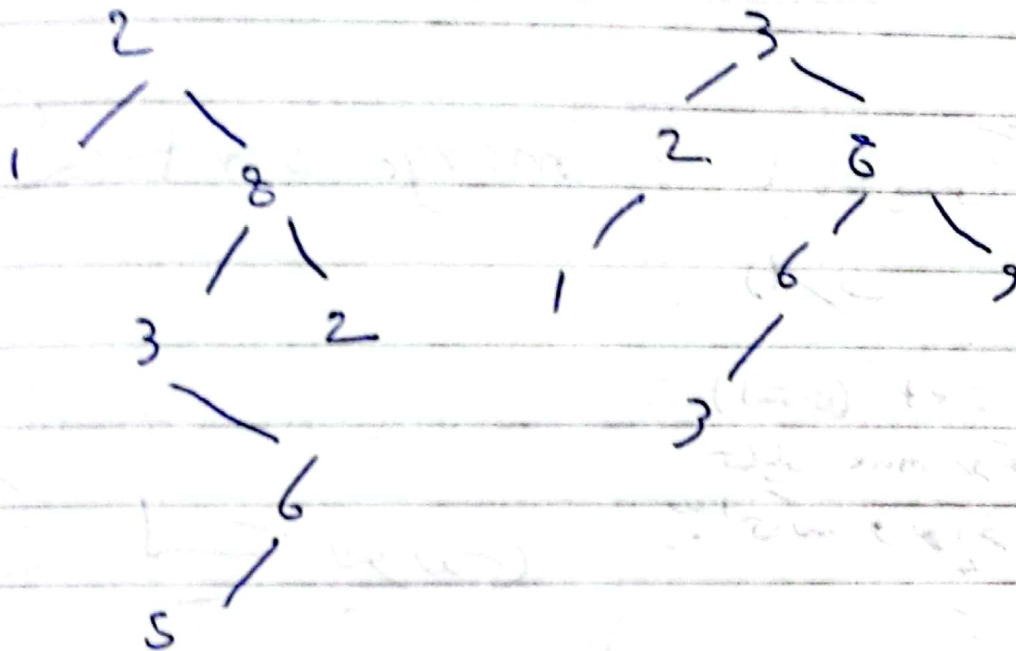
حبابی هم توهم به مقایسه هر دو آراء  
 کنار هم رو با هم مقایسه می کنند مثلاً خانه  
 1 با 2 به خانه 2 با 3 و 3 با 4 ...  
 { تراز می کنند اول و آخر }  
 تا صورت بیاید

لذاً merge sort مثل همدن روشی اول است

چون نوی هر کدام زنجیره می باشد. اندیس و پیوندی  
 نهایی گت بهترین

سورج کردن  
 Binary tree  
 Nike

2, 8, 1, 3, 6, 5, 9



این : روش مرتب سازی حبابی را

در نظر بگیرید - همچنین روش مرتب سازی

حبابی (Bubble sort) . این ۲ روش مرتب

CPU

سازی را در حالت load روی

با یلفر مقایسه کنید .