Modify the induction conclusion

$$T(n) \leq 2T\left(\frac{n}{2}\right) + \sqrt{n}$$

Guess $T(n) \leq c \cdot n$

$$\begin{aligned} T(n) &\leq 2 \cdot T\left(\frac{n}{2}\right) + \sqrt{n} \\ &\leq 2 \cdot c \cdot \frac{n}{2} + \sqrt{n} \\ &= cn + \underline{\sqrt{n}} \end{aligned}$$

Lemma: $T(n) \leq c \cdot n - 3 \cdot \sqrt{n}$

proof: Omit base case

$$\begin{aligned} T(n) &\leq 2 \cdot T\left(\frac{n}{2}\right) + \sqrt{n} \\ &\leq 2 \cdot \left(c \cdot \frac{n}{2} - 3\sqrt{\frac{n}{2}}\right) + \sqrt{n} \\ &= c \cdot n \boxed{- 3\sqrt{2}\sqrt{n} + \sqrt{n}} \rightarrow -(3\sqrt{2} - 1)n \\ &\leq c \cdot n - 3 \cdot \sqrt{n} \end{aligned}$$

$= 1.414 \times 3 - 1 \approx 3.2 > 3$

Variable Subsitutions

$$T(n) = 2 \cdot T(\sqrt{n}) + \log n$$

$$m = \log n$$

$$\begin{aligned} S(m) &= T(2^m) \\ &= 2 \cdot T(2^{m/2}) + m \\ &= 2 \cdot S(m/2) + m \end{aligned}$$

$$S(m) = O(m \cdot \log m)$$

$$\begin{aligned} T(n) &= S(\log_2 n) \\ &= O(\log_2 n \log \log n) \end{aligned}$$

Master's Theorem

$\underline{n} \to \underline{a}$ smaller problems of size n/b
fix up the solution with $\underline{n^c}$

$T(n) = a \cdot T(n/b) + n^c$
if $a \geq 1$, $b \geq 1$, $c \geq 0$, and

Then:

$$T(n) \begin{cases} \Theta(n^c), \text{ if } c > \log_b a \\ \Theta(n^c \cdot \log n), \text{ if } c = \log_b a \\ \Theta(n^{\log_b a}), \text{ if } c < \log_b a \end{cases}$$

first example: $a = b = 2$, $c = \frac{1}{2}$, $c < \log_b a$, $O(n)$

---

proof of case 1:
To prove $T(n) \leq \gamma \cdot n^c$ for $\gamma > 0$

$$\begin{aligned} T(n) &= a \cdot T(\tfrac{n}{b}) + n^c \\ &\leq a \cdot \gamma \cdot (\tfrac{n}{b})^c + n^c \\ &= (\tfrac{a}{b^c} \cdot \gamma + 1) \cdot n^c \end{aligned}$$

To prove the theorem, we only need to choose $\gamma > 0$, s.t.

$$\frac{a}{b^c} \cdot \gamma + 1 = \gamma$$

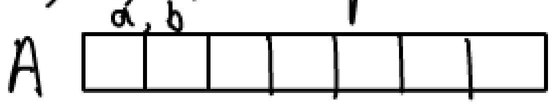
$$\gamma = \frac{1}{1 - \frac{a}{b^c}} > 0$$

$\hookrightarrow c > \log_b a$

Counting inversion

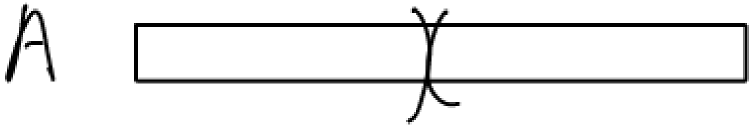
$1, 2, 3, \dots, n$

$i_1, i_2, \dots, i_n$ ← permutation

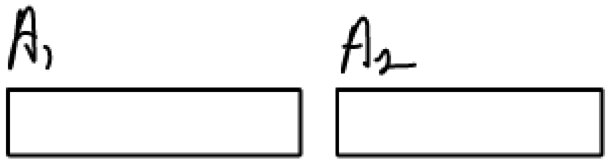


if $j < k$, and $i_j > i_k$

then (j, k) is an inversion



inv(A)

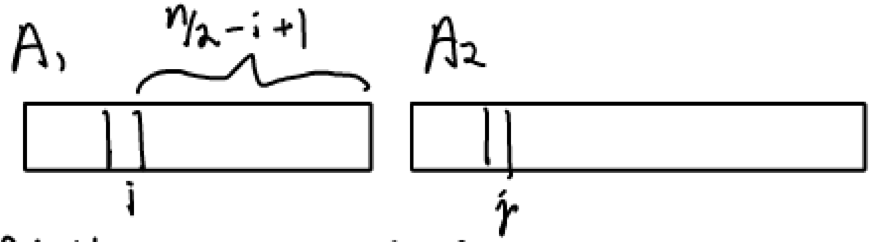


$Inv(A_1) + Inv(A_2) + Inv(A_1, A_2) = inv(A)$

$\frac{n}{2} \times \frac{n}{2}$

$T(n) = 2T(n/2) + n^2/4$

$= O(n^2)$



A_1, A_2 sorted

```
if (A1[i] ≤ A2[j])
    i++;
else
    counter += n/2 - i + 1;
    j++;
```

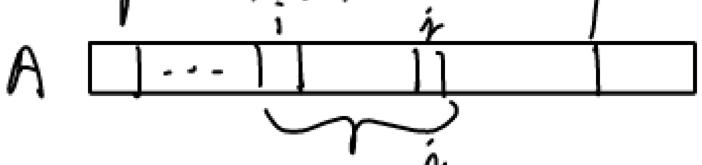
```
counter = 0
while (i < n/2 || j < n/2) {

}
```

So if we sort:

$$T(n) = 2\cdot T(\tfrac{n}{2}) + n$$
$$= O(n \cdot \log n)$$

Example: Max Subarray

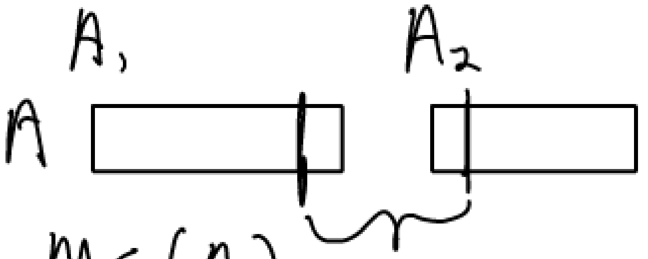


$$c_{i,j} = \sum_{k=i}^{j} A[k]$$

Trivial I: $O(n^3)$

Trivial II: $O(n^2)$

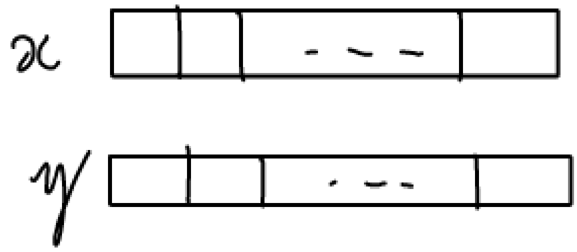
for $i=1..n$
  for $j=i+1..n$
    $c_{i,j} = c_{i,j-1} + A[j]$



$$MS(A) = \max \begin{cases} MS(A_1) \\ MS(A_2) \\ \text{maxsuffix}(A_1) + \text{maxprefix}(A_2) \end{cases}$$

$$T(n) = 2T(\tfrac{n}{2}) + O(n)$$
$$= O(n \log n)$$

Integer Multiplication



$O(n^2) \longrightarrow$

$$\begin{array}{r} 1\,2\,3 \\ \times \;\; 3\,2\,1 \\ \hline 1\,2\,3 \\ 2\,4\,6\;\; \\ 3\,6\,9\;\;\;\; \\ \hline 3\,9\,5\,8\,3 \end{array}$$

$m = \frac{n}{2}$

$x = 2^m \cdot x_1 + x_2$

$y = 2^m \cdot y_1 + y_2$

$x \cdot y = 2^n \cdot x_1 y_1 + 2^m (x_1 y_2 + x_2 y_1) + x_2 y_2$

$T(n) = 4 \cdot T(\frac{n}{2}) + O(n)$

$a = 4$
$b = 2$
$c = 1$

$c < \log_b a$ $\quad n^{\log_b a} = n^2$ $\quad$ lol