Algorithms

* review runtimes for access of different data structures *

```
01  closestpair(p_1, ..., p_n) : array of 2D points
02        best1 = p_1
03  O(1)  best2 = p_2
04        bestdist = dist(p_1,p_2)
05  →     for i = 1 to n
06            for j = 1 to n
07                newdist = dist(p_i,p_j)   → any two points
08    n times    n times    if (i ≠ j and newdist < bestdist)   O(1)
09                    best1 = p_i
10                    best2 = p_j    O(1)
11                    bestdist = newdist
12  O(1)  return (best1, best2)
```

02-04: $O(1)$

05-11: $O(n \cdot n \cdot 1) = O(n^2)$

12: $O(1)$

$O(1+n^2+1) = O(n^2+2) = O(n^2)$

```
mergesort(L = a_1, a_2, ..., a_n: list of real numbers)
    if (n = 1) then return L
    else
        m = ⌊n/2⌋
        L_1 = (a_1, a_2, ..., a_m)
        L_2 = (a_{m+1}, a_{m+2}, ..., a_n)
        return merge(mergesort(L_1),mergesort(L_2))
```

1 2 $O(1)$ $O(n)$ 3 4 $O(n)$

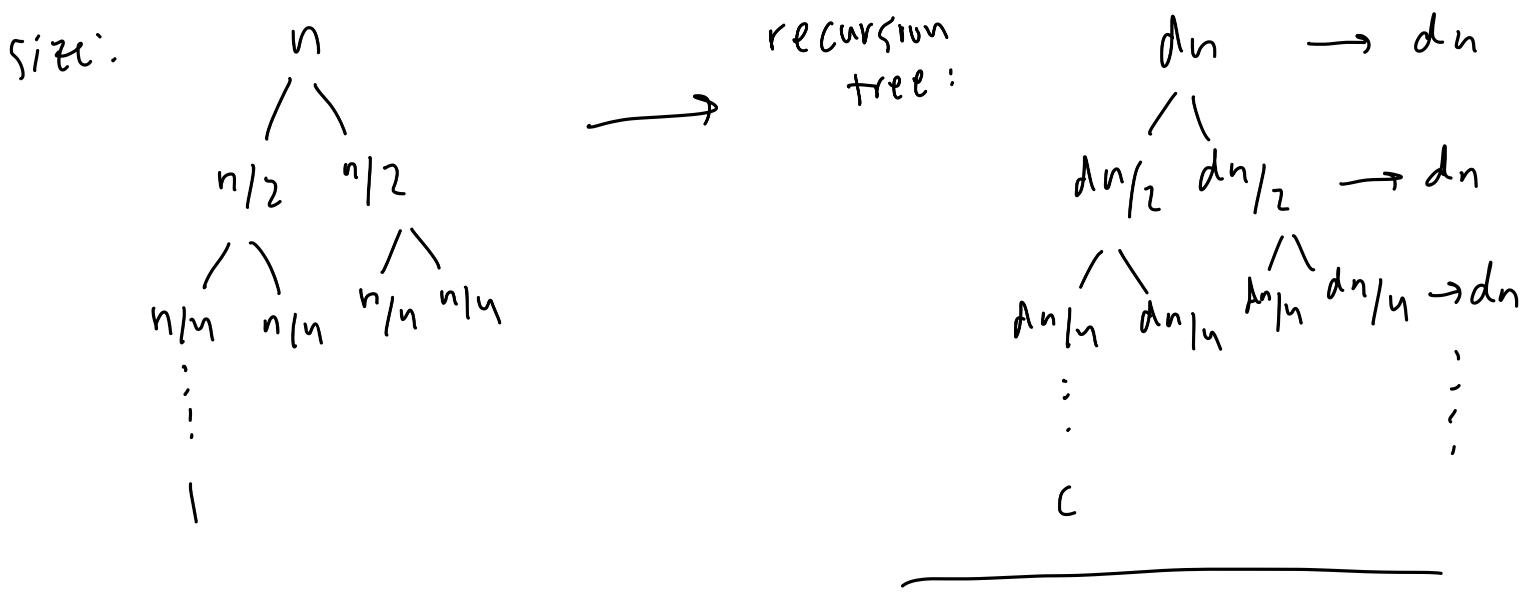
$T(1) = c$

$T(n) = 2T(n/2) + \underbrace{O(n)}_{\text{extra work}}$

$T(n) = 2T(n/2) + dn$

$T(1) = c$

$T(n) = 2T(n/2) + dn$



total work: $dn(\log_2 n) + c(2^{\log_2 n})$

$dn \cdot \log_2 n + cn$

$O(n\log_2 n) = \underline{O(n\log n)} \rightarrow O(n^2)$

$\Theta(n\log n)$