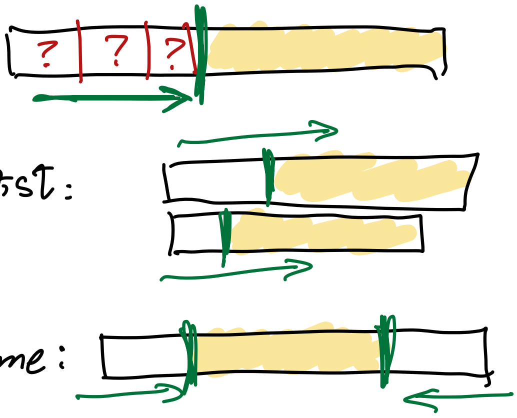
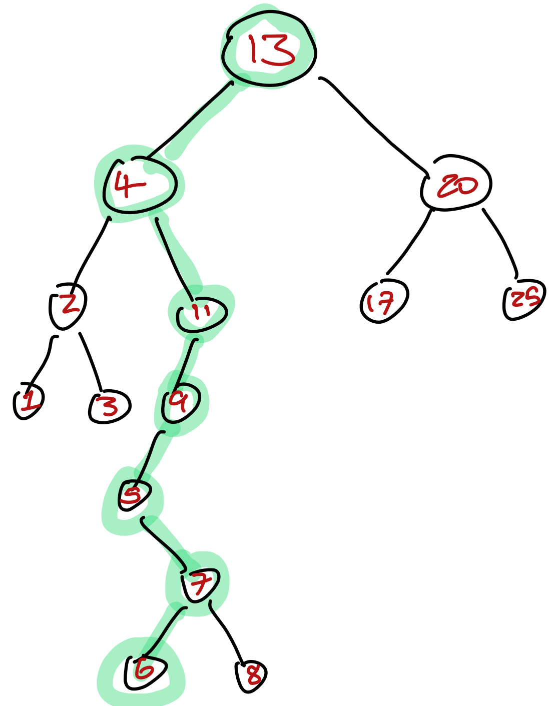
Sequences:

Edit Dist:

Palindrome:

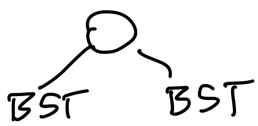




Binary Search Tree

— nothing (NULL)

—



If we know how often we look for each item

what is the best BST?

Keys = $1, 2, \ldots, n$

Frequencies $f[1..n]$ ← INPUT

$f[i]$ = # times we search for key i.

$$Cost(T, f) = \sum_{i=1}^{n} f[i] \cdot depth(i, T)$$

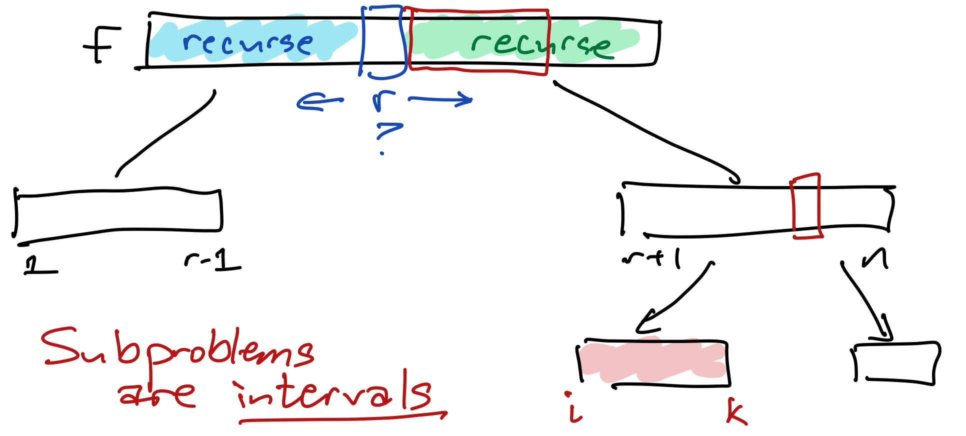
$$= \boxed{\sum_{i} f[i] \cdot \#\text{ancestors of } i \text{ in } T}$$

$$= \sum_{i} f[i] \cdot \left(\begin{array}{l} 1 + \#\text{ancestors of } i \text{ in } left(T) \\ + \#\text{anc of } i \text{ in } right(T) \end{array} \right)$$

$$= \boxed{\sum_{i=1}^{r-1} f[i] \cdot \#\text{anc in left}} + \sum_{i=1}^{n} f[i] + \boxed{\sum_{i=r+1}^{n} f[i] \cdot \#\text{anc in right}}$$

$$Cost(T, f[1..n]) = \sum_{i=1}^{n} f[i] + Cost(left(T), f[1..r-1]) + Cost(right(T), f[r+1..n])$$

$r = root(T)$



Subproblems are intervals

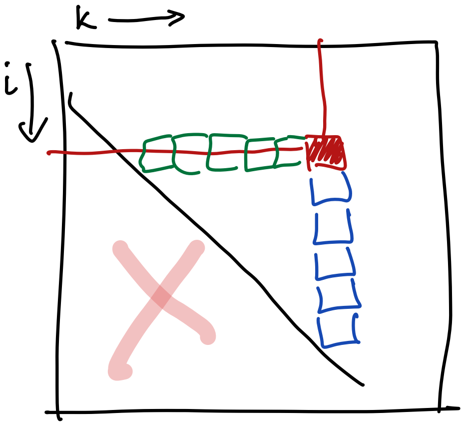
OptCost(i,k) = total cost of opt. BST for frequencies F[i..k]

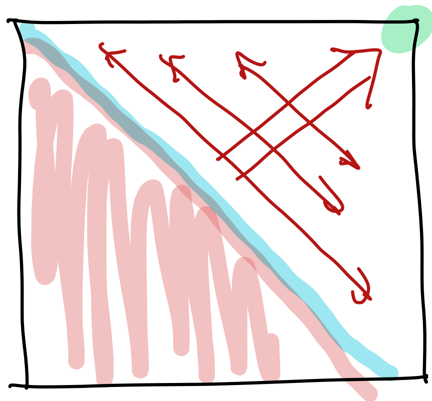
$$OptCost(i,k) = \begin{cases} 0 & \text{if } i > k \\ \underbrace{\sum_{j=i}^{k} f[j]}_{\text{\#times we touch the root}} + \underbrace{\min_{i \le r \le k}}_{\text{choose root } r} \left\{ \begin{array}{l} \overbrace{OptCost(i, r-1)}^{\text{left subtree}} \\ \quad + \underbrace{OptCost(r+1, k)}_{\text{right subtree}} \end{array} \right\} & \text{otherwise} \end{cases}$$

$$F[i,k] = \sum_{j=i}^{k} f[j] = \begin{cases} 0 & \text{if } i > k \\ F[i,k-1] + F(k) & \text{o/w} \end{cases}$$

```
InitF(f[1..n]):
  for i ← 1 to n
      F[i, i−1] ← 0
      for k ← i to n
          F[i, k] ← F[i, k−1] + f[k]
```

$$OptCost(i,k) = \begin{cases} 0 & \text{if } i > k \\ F[i,k] + \min_{i \le r \le k} \left\{ \begin{array}{l} OptCost(i, r-1) \\ \quad + OptCost(r+1, k) \end{array} \right\} & \text{otherwise} \end{cases}$$

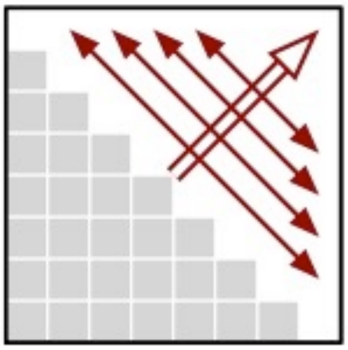


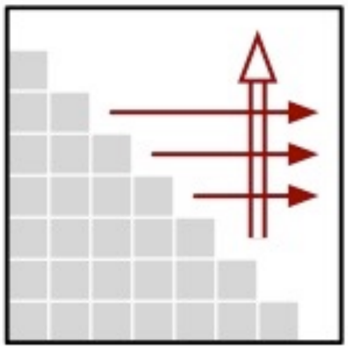


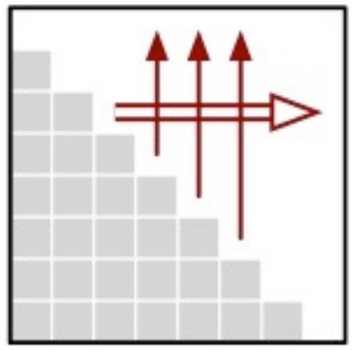
```
ComputeOptCost(i, k):
    OptCost[i, k] ← ∞
    for r ← i to k
        tmp ← OptCost[i, r − 1] + OptCost[r + 1, k]
        if OptCost[i, k] > tmp
            OptCost[i, k] ← tmp
    OptCost[i, k] ← OptCost[i, k] + F[i, k]
```

```
OptimalBST(f[1..n]):
    InitF(f[1..n]) ⇐ O(n^2) time
    for i ← 1 to n + 1
        OptCost[i, i − 1] ← 0
    for d ← 0 to n − 1
        for i ← 1 to n − d          ⟨⟨. . . or whatever⟩⟩
            ComputeOptCost(i, i + d) ⇐ O(n) time
    return OptCost[1, n]
```

$O(n^3)$ time







```
OptimalBST(f[1..n]):
  InitF(f[1..n])
  for i ← 1 to n+1
      OptCost[i, i−1] ← 0
  for d ← 0 to n−1
      for i ← 1 to n−d        ⟨⟨... or whatever⟩⟩
          ComputeOptCost(i, i+d)
  return OptCost[1, n]
```

```
OptimalBST2(f[1..n]):
  InitF(f[1..n])
  for i ← n+1 downto 1
      OptCost[i, i−1] ← 0
      for j ← i to n
          ComputeOptCost(i, j)
  return OptCost[1, n]
```

```
OptimalBST3(f[1..n]):
  InitF(f[1..n])
  for j ← 0 to n+1
      OptCost[j+1, j] ← 0
      for i ← j downto 1
          ComputeOptCost(i, j)
  return OptCost[1, n]
```

$$OptCost(i,k) = \begin{cases} 0 & \text{if } i > k \\ F[i,k] + \min\limits_{i \le r \le k} \left\{ \begin{array}{l} OptCost(i, r-1) \\ \quad + OptCost(r+1, k) \end{array} \right\} & \text{otherwise} \end{cases}$$

2 input params $\Rightarrow O(n^2)$ space

3 vars on right $\Rightarrow O(n^3)$ time