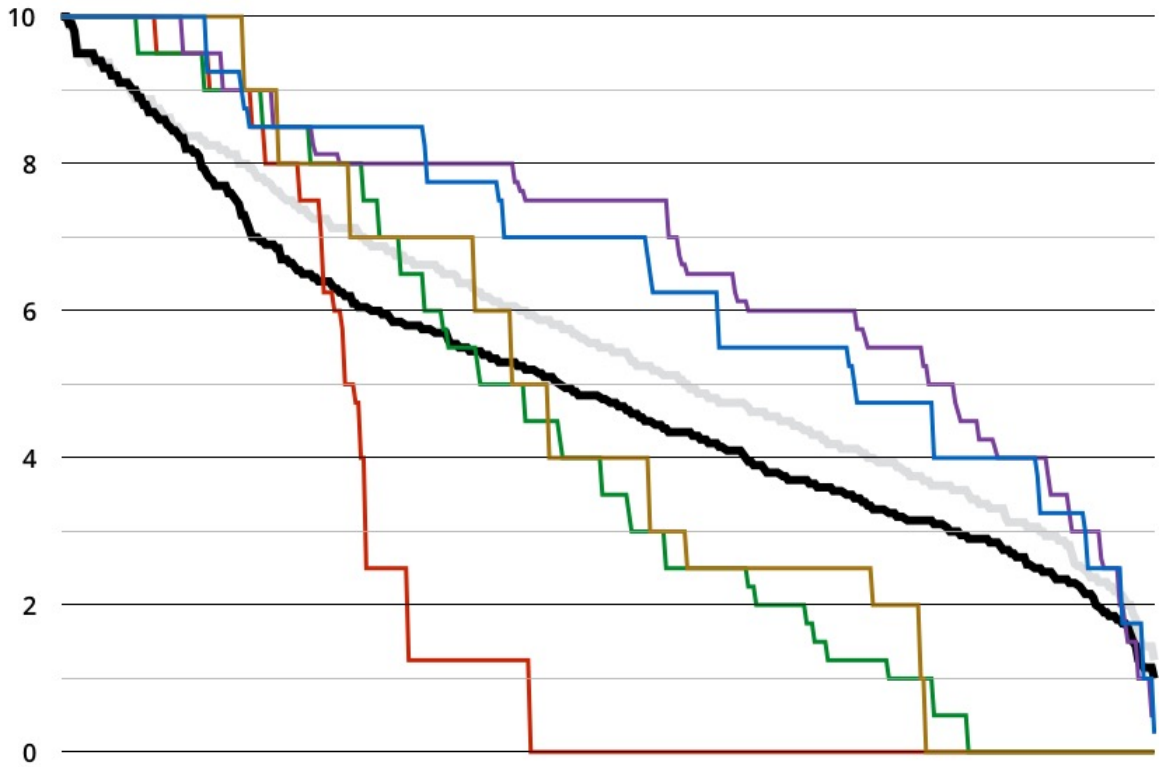
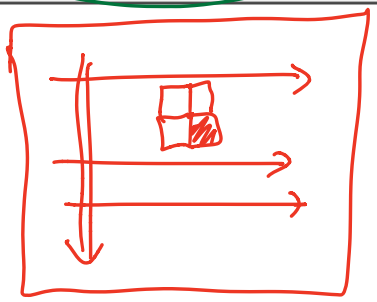

10
8
6
4
2
0

Given $A[1..m]$ $B[1..n]$

min # ins/del/reps to change A into B

$Edit(i,j)$ = edit distance $A[1..i]$ and $B[1..j]$

$$Edit(i,j) = \begin{cases} i & \text{if } j = 0 \\ j & \text{if } i = 0 \\ \min \begin{cases} Edit(i-1,j)+1, & \text{del} \\ Edit(i,j-1)+1, & \text{ins} \\ Edit(i-1,j-1)+[A[i] \neq B[j]] & \text{rep} \end{cases} & \text{otherwise} \end{cases}$$



```
EditDistance(A[1..m], B[1..n]):
  for j ← 0 to n
      Edit[0, j] ← j

  for i ← 1 to m
      Edit[i, 0] ← i
      for j ← 1 to n
          if A[i] = B[j]
              Edit[i, j] ← min {Edit[i−1, j]+1, Edit[i, j−1]+1, Edit[i−1, j−1]}
          else
              Edit[i, j] ← min {Edit[i−1, j]+1, Edit[i, j−1]+1, Edit[i−1, j−1]+1}
  return Edit[m, n]
```

$O(mn)$ time

		A	L	G	O	R	I	T	H	M
	0	→1	→2	→3	→4	→5	→6	→7	→8	→9
A	1	0	→1	→2	→3	→4	→5	→6	→7	→8
L	2	1	0	→1	→2	→3	→4	→5	→6	→7
T	3	2	1	1	→2	→3	→4	4	→5	→6
R	4	3	2	2	2	2	→3	→4	→5	→6
U	5	4	3	3	3	3	3	→4	→5	→6
I	6	5	4	4	4	4	3	→4	→5	→6
S	7	6	5	5	5	5	4	4	5	6
T	8	7	6	6	6	6	5	4	→5	→6
I	9	8	7	7	7	7	6	5	5	→6
C	10	9	8	8	8	8	7	6	6	6

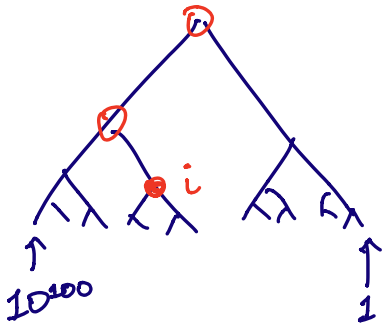
A	L	G	O	R	I		T	H	M
A	L	T	R	U	I	S	T	I	C

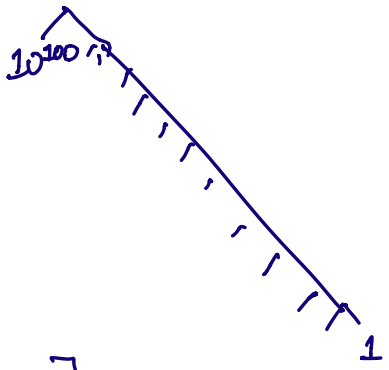
A	L	G	O	R		I		T	H	M
A	L		T	R	U	I	S	T	I	C

A	L	G	O	R		I		T	H	M
A	L	T		R	U	I	S	T	I	C

Optimal BST problem

Given: search keys $A[1..n]$ sorted
Frequencies $F[1..n]$



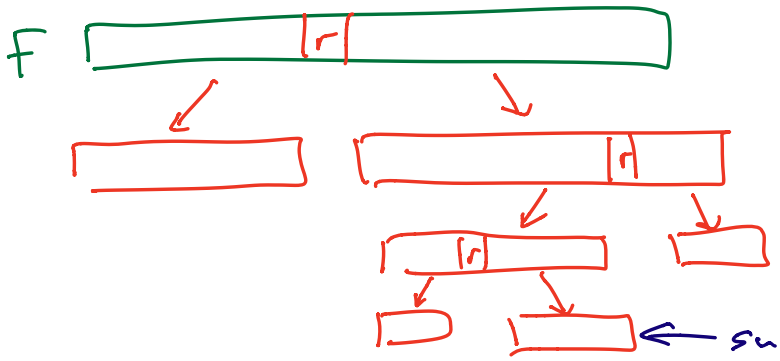


Build a BST for $A[1..n]$
that minimizes

$$Cost(T, F[1..n]) = \sum_{i=1}^{n} F[i] \cdot (\#\text{ancestors of } i \text{ in } T)$$

$$= \underbrace{\sum_{i=1}^{n} F[i]}_{\text{root}} + \sum_{i=1}^{r-1} F[i] \cdot (\#\text{ancestors of } i \text{ in } left(T)) + \sum_{i=r+1}^{n} F[i] \cdot (\#\text{ancestors of } i \text{ in } right(T))$$

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



OptCost(i,k) = Cost of optimal BST for keys A[i..k] and frequencies f[i..k]

$$\text{OptCost}(i,k) = \begin{cases} 0 & \text{if } i > k \\ \sum_{r=i}^{k} f[r] + \min_{i \le r \le k} \left(\text{OptCost}(i, r-1) + \text{OptCost}(r+1, k) \right) & \end{cases}$$

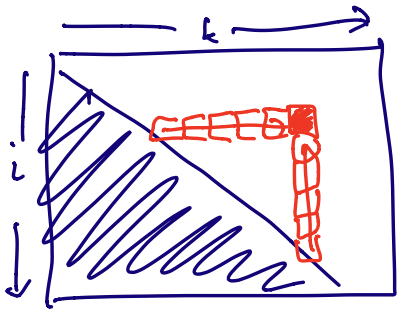
Subproblems $1 \le i \le n+1$ $0 \le k \le n$ $i \le k+1$

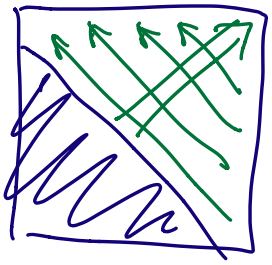
Memoize

Dependencies

Order

Time $O(n^3)$



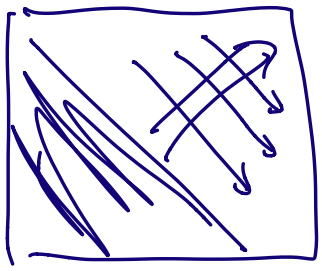


$$F[i,k] = \sum_{j=1}^{k} F[j]$$

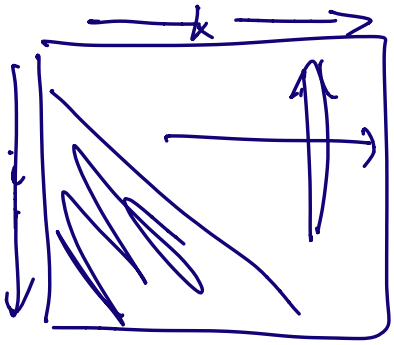
```
INITF(f[1..n]):
  for i ← 1 to n
      F[i, i−1] ← 0
      for k ← i to n
          F[i,k] ← F[i,k−1] + f[k]
```

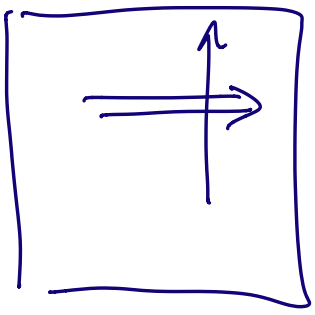
$$OptCost(i,k) = F[i,k] + \min_{i \le r \le k} \left(\begin{matrix} OptCost(i,r-1) \\ + \\ OptCost(r+1,k) \end{matrix} \right)$$

```
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])
  for i ← 1 to n+1
      OptCost[i, i−1] ← 0
  for d ← 0 to n−1
      for i ← 1 to n−d
          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]
```

Memoize into the tree itself

eval in post order

```
MaximumIndSetSize(v):
  withoutv ← 0
  for each child w of v
      withoutv ← withoutv + MaximumIndSetSize(w)
  withv ← 1
  for each grandchild x of v
      withv ← withv + x.MIS
  v.MIS ← max{withv, withoutv}
  return v.MIS
```

```
MaximumIndSetSize(v):
  v.MISno ← 0
  v.MISyes ← 1
  for each child w of v
      v.MISno ← v.MISno + MaximumIndSetSize(w)
      v.MISyes ← v.MISyes + w.MISno
  return max{v.MISyes, v.MISno}
```