COP4516 4/1/2025

1. Matrix Chain Multiplication
2. World Series Problem
3. Edit Distance + HSPT
4. Traveling Salesmen DP or Driving Lanes?

MCM

A = 2×5
B = 5×3
C = 3×4
D = 4×1

ABCD will be a 2×1 matrix

Cost(AB) = $A_{rows} \times A_{cols} \times B_{cols}$

ABCD = A | (BCD)

(AB) | (CD)

(ABC) | D

CYK Algorithm

OUTLINE

A[0, 1, 2, ... n-1]
start end

```
f(start, end) {
  if (memo[start][end] != -1) return memo[start][end];
  if (start >= end) return 0;
  res = MAX
  for (int mid = start; mid < end; mid++)
    res = min(res, f(start, mid) +
                   f(mid+1, end) +
                   cost(start, mid, end))
  return memo[start][end] = res;
}
```

Played at most $2n-1$ game. 1st team to n games wins series. Probability of winning a games for team A is p.

$p(a,b)$ = probability after $a+b$ games that team A has won a times and team B has won b ~~tea~~ times.

$$p(a,b) = p(a-1,b)*p + p(a,b-1)*(1-p)$$

- $p(a-1,b)$: Prob of $a-1$ w/ A & b w/ B in first $a+b-1$ games
- p: A wins last
- $p(a,b-1)$: Prob of a w/ A & $b-1$ w/ B in first $a+b-1$ games
- $(1-p)$: B wins last

Edit Distance

starting string: hello
end string: bedlow

} goal transform start string → end string

min cost

moves: a) insert a char
b) del a char
c) change a char

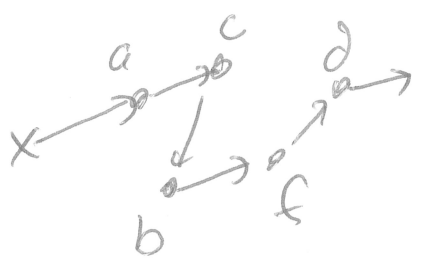
LCS $f(i,j) =$ ~~1+~~ $f(i-1,j-1)$ match
$= \min($ ~~1+~~ $f(i-1,j), $ ~~1+~~ $f(i,j-1))$

EDITD $f(i,j) = f(i-1,j-1)$ match
$= \min(1+f(i-1,j), 1+f(i,j-1), 1+f(i-1,j-1))$

Traveling Salesman DP

Input: n cities, distances btw all pairs

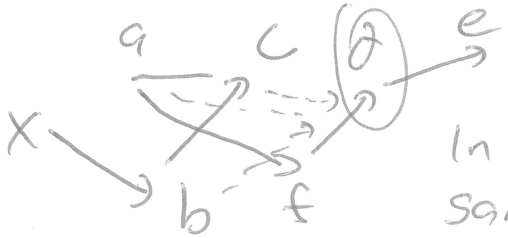
Goal: Start ~~xx~~ at home (diff locations)
visit each location exactly once and a end anywhere traveling min distance possible



50
a, c, b, f, d, e (my path)

CS1 SOL: Do all n permutations + take best ans $O(n \times n!)$

alt path



62
b, c, a, f, d, e

In these 2 paths we've visited same set of vertices + ended at the same place

Define $f(abcf, f)$ = min cost visiting subset $\{a, b, c, f\}$ ending at f.

$$f(abcdf, d) = \min\begin{cases} f(abcf, f) + fd, & (\text{get to } d \text{ from } f) \\ f(abcf, c) + cd, & (\text{get to } d \text{ from } c) \\ f(abcf, b) + bd, & (\text{get to } d \text{ from } b) \\ f(abcf, a) + ad & (\text{get to } d \text{ from } a) \end{cases}$$

subsets size n ↓ — ↑ bitmask

```
dp[1<<n][n]
// MASK SUBSET
for (int mask=1; mask<(1<<n); mask++) {
    // j last item visited
    for (int j=0; j<n; j++) {
        if ((mask & (1<<j)) == 0)
            continue
        int lastmask = mask - (1<<j)
        res = INF;
        // k is last item in lastmask
        for (int k=0; k<n; k++) {
            if ((lastmask & (1<<k)) == 0)
                continue;
            res = min(res, dp[lastmask][k] +
                        g[k][j]);
        }
        dp[mask][j] = res;
    }
}
```