# CS601: Software Development for Scientific Computing

Autumn 2024

Week16 : Dynamic Programming (DP) Problems

# Course Progress..

- **Pic source: the Parallel Computing Laboratory at U.C. Berkeley: A Research Agenda Based on the Berkeley View (2008)**

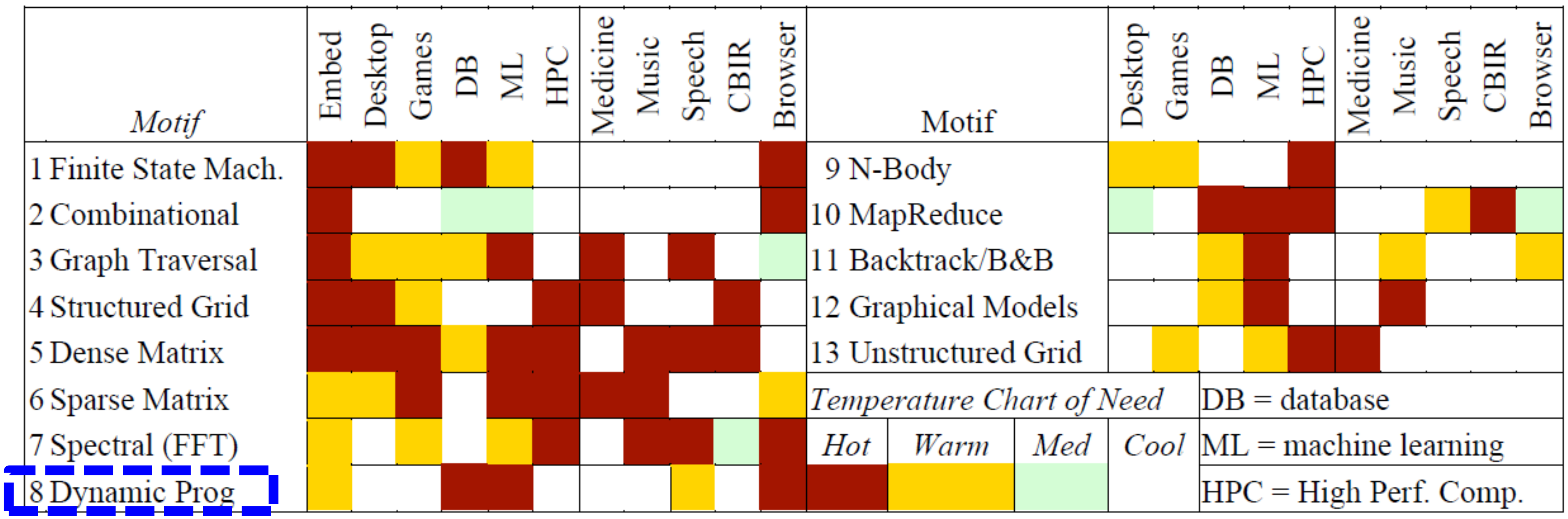



**Figure 4. Temperature Chart of the 13 Motifs.** It shows their importance to each of the original six application areas and then how important each one is to the five compelling applications of Section 3.1. More details on the motifs can be found in (Asanovic, Bodik et al. 2006).

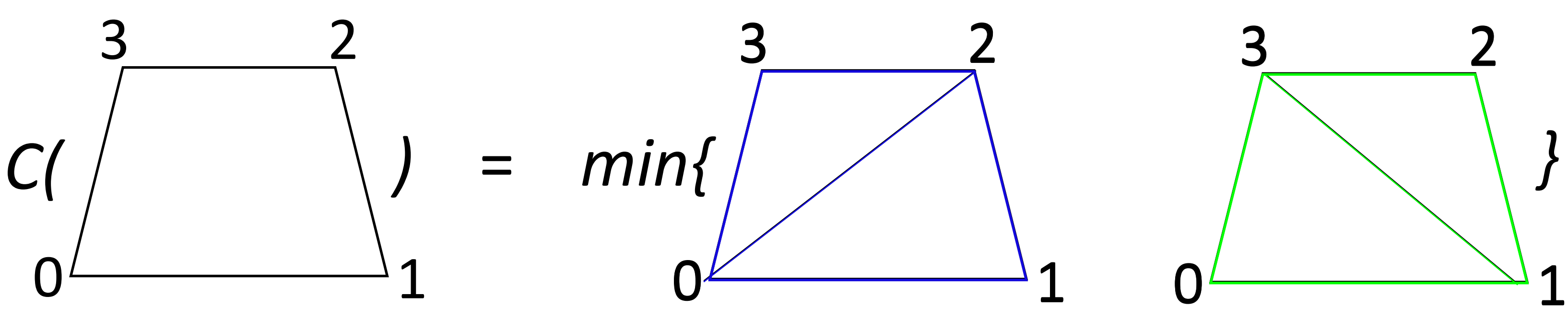
Next..

# Example – Minimum Weight Triangulation

## Example of ‘Parenthesis’ problem

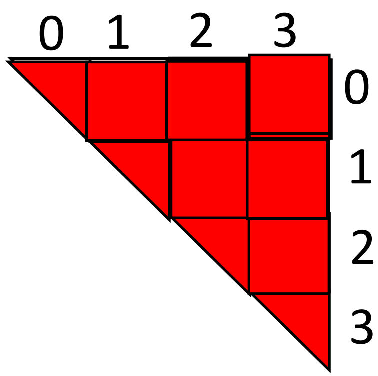
- Used in Computational Geometry, Finite Element Methods
  - Objective: Triangulate a convex polygon such that edges do not intersect AND the sum of the edge lengths is minimized

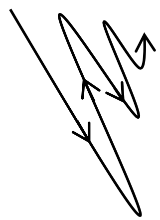
$$C(i,j) = \begin{cases} min\big(\ C(i,j),\ \min\limits_{i<k<j} C(i,k) + C(k,j) + W(i,k,j)\big) & \\ 0 & j \leq\ i+1 \\ Given\ W(i,j,k) & \end{cases}$$




C(0,3) = min{C(0,2) + C(2,3) + $\triangle^{le}$ (0,2,3), C(0,1)+C(1,3)+$\triangle^{le}$ (0,1,3)}

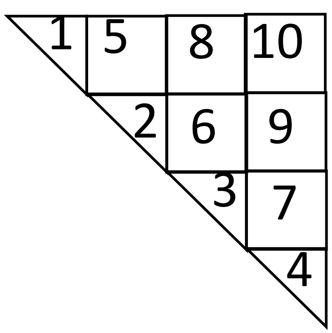
# Implementation - ‘standard’ (iterative) approach



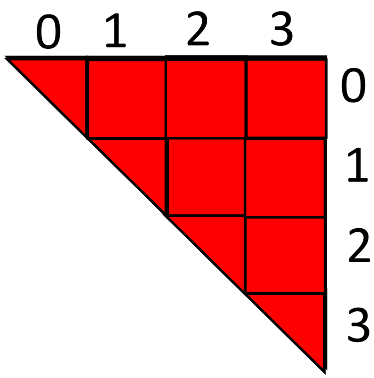
```
Cost(n){
  table[n][n]; //n is number of vertices
  for g ← 1 to n-1 do
    for i ← 0 to n-g do
      j ← i + g;
      table[i][j] ← INFINITY;
      for k ← i+1 to j–1 do
        res ← table[i][k] + table[k][j] + Weight(i,k,j)
        if res < table[i][j] then
          table[i][j] ← res;
}
```

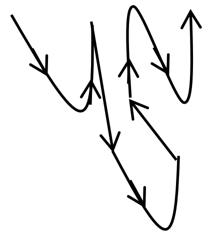
**Iterative formulation**

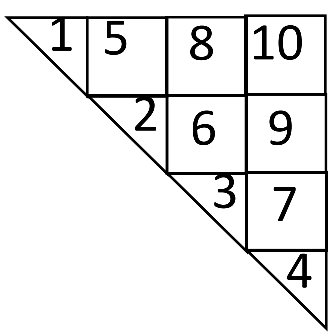



- Computing sequence: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

# Implementation - recursive formulation






A(X,X,X)

*A(X00, X00, X00); A(X11, X11, X11);*
*B(X01, X00, X11);*

B(Z,U,V)

No branch statements

*C(Z00, U01, Z10); C(Z11, Z10, V01);*

Predictable, input-independent computation (top-left cell to top-right cell)

*B(Z01, U00, V11);*

X00, X01, and X10 are quadrants within X, method A's parameter.

C (T,R,S)

*C(T00, R00, S00); C(T01, R00, S01);*
*C(T10, R10, S00); C(T11, R10, S01);*
*C(T00, R01, S10); C(T01, R01, S11);*
*C(T10, R11, S10); C(T11, R11, S11);*

```
main(){
 X[4][4];
 A(X,X,X);
}
```

Recursive sequence: 1, 2, 5, 3, 4, 7, 6, 8, 9, 8, 9, 10, 10, 10

Iterative sequence: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

# Example of ‘Parenthesis’ problems

- Matrix Chain Multiplication
- Minimum Weight Triangulation

$$D[i, j] = \min_{i<r<j} \{D[i, r] + D[r, j] + w(i, r, j)\}$$
$$\text{for} \quad 0 \leq i < j \leq n.$$

# Example of '1D' problems

- Rod cutting problem
  - Refer to Chapter 15, Section 15.1 in "Introduction to Algorithms", 3rd Ed. T. E. Cormen, C.E. Leiserson, R. L. Rivest, Clifford Stein.

- Least weight subsequence problem:

$$D[j] = \min_{0 \le i < j} \{D[i] + w(i, j)\} \quad \text{for } 1 \le j \le n.$$

# Example of 'Gap' Problems

- [Smith-Waterman]

- [All Pairs Shortest Path by Floyd-Warshall]

$$\text{shortestPath}(i, j, k) =$$
$$\min\Big(\text{shortestPath}(i, j, k-1),$$
$$\text{shortestPath}(i, k, k-1) + \text{shortestPath}(k, j, k-1)\Big)$$

# Example of 'RNA' Problems

- Smith-Waterman (RNA secondary structure prediction without multiple loops)

- Nussinov's (algorithms for loop matching)

$$D[i,j] = \min_{\substack{0 \le p < i \\ 0 \le q < j}} \{D[p,q] + w(p,q,i,j)\} \quad \text{for} \quad 1 \le i,j \le n. \tag{4}$$