Dynamic Programming

- Weighted Interval Scheduling
- Given a collection of intervals \( I_1, \ldots, I_n \) with weights \( w_1, \ldots, w_n \), choose a maximum weight set of non-overlapping intervals

Recursive Algorithm

Intervals sorted by finish time
\( p[i] \) is the index of the last interval which finishes before \( i \) starts

Optimality Condition

- \( \text{Opt}[j] \) is the maximum weight independent set of intervals \( I_1, I_2, \ldots, I_j \)

Algorithm

\[
\text{MaxValue}(j) = \\
\text{if } j = 0 \text{ return } 0 \\
\text{else return } \max( \text{MaxValue}(j-1), w_j + \text{MaxValue}(p[j]))
\]

Run time

- What is the worst case run time of \( \text{MaxValue} \)
- Design a worst case input
A better algorithm

MaxValue(j) =
if j = 0 return 0;
else if M[ j ] != -1 return M[ j ];
else
    M[ j ] = max(MaxValue(j-1),w_{j} + MaxValue(p[ j ]));
    return M[ j ];

Iterative Algorithm

Express the MaxValue algorithm as an
iterative algorithm

MaxValue {

}

Fill in the array with the Opt values

Opt[ j ] = max (Opt[ j – 1], w_{j} + Opt[ p[ j ] ])

Computing the solution

Opt[ j ] = max (Opt[ j – 1], w_{j} + Opt[ p[ j ] ])

Dynamic Programming

• The most important algorithmic technique
covered in CSE 421

• Key ideas
  – Express solution in terms of a polynomial
    number of sub problems
  – Order sub problems to avoid recomputation

Optimal linear interpolation

Error = \sum(y_{i} – ax_{i} – b)^2
What is the optimal linear interpolation with three line segments

What is the optimal linear interpolation with two line segments

What is the optimal linear interpolation with \( n \) line segments

### Notation
- Points \( p_1, p_2, \ldots, p_n \) ordered by \( x \)-coordinate \( (p_i = (x_i, y_i)) \)
- \( E_{i,j} \) is the least squares error for the optimal line interpolating \( p_i, \ldots, p_j \)

### Optimal interpolation with two segments
- Give an equation for the optimal interpolation of \( p_1, \ldots, p_n \) with two line segments

- \( E_{i,j} \) is the least squares error for the optimal line interpolating \( p_i, \ldots, p_j \)

### Optimal interpolation with \( k \) segments
- Optimal segmentation with three segments
  - \( \text{Min}_{i,j} \{E_{i,j} + E_{j,n} + E_{i,n} \} \)
  - \( O(n^2) \) combinations considered
- Generalization to \( k \) segments leads to considering \( O(n^{k-1}) \) combinations
**Optimal sub-solution property**

Optimal solution with \( k \) segments extends an optimal solution of \( k-1 \) segments on a smaller problem.

---

**Optimal multi-segment interpolation**

Compute \( \text{Opt}[k, j] \) for \( 0 < k < j < n \)

for \( j := 1 \) to \( n \)

\[ \text{Opt}[1, j] = E_{1,j} \]

for \( k := 2 \) to \( n-1 \)

for \( j := 2 \) to \( n \)

\[ t := E_{1,j} \]

for \( i := 1 \) to \( j-1 \)

\[ t = \min(t, \text{Opt}[k-1, i] + E_{i,j}) \]

\[ \text{Opt}[k, j] = t \]