Design and Analysis of Algorithms

6.3 Dynamic Programming Exercises

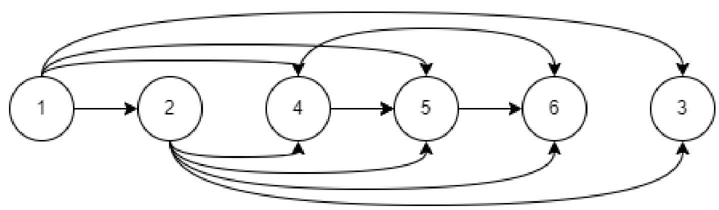
Daniel Shannon

May 16th, 2022

6.3.2

- You have a sequence of numbers $x_1, x_2, x_3, ..., x_n$
- Find the contguous subsequence $[x_i, ..., x_j]$ with the greatest sum
- Not allowed to skip elements!
- Use dynamic programming to find an $O(n)$ solution

My Answer



example linearized DAG

Similar to largest increasing subsequence, but we track the sum instead of the length.

$prev(0) = x_1$
for all $j = 1, 2, ...n$, in linearized order **do**
  $sum_j {+}{=} max\{S[j] + x_j : (i, j) \in E\}$
  if $S[j] < sum_j$ **then**
    $S[j] = sum_j$
    $prev(j) \cdot (i, j)$ concat
  end if
end for

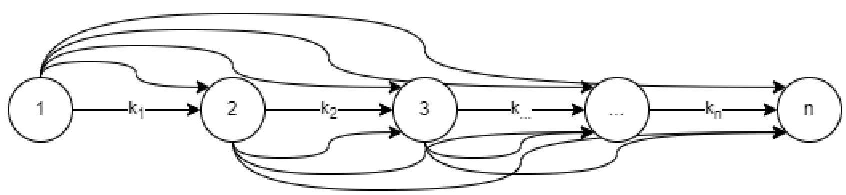
Answer

$S[j]$ maximum sum of a cont subsequence ending at index j
$S[j] = max\{A[j], S(j-1) + A[j]\}$
$S[0] = 0$
for all $j = 1, 2, ...n$, in linearized order **do**
  $S[j] = max\{A[j], S(j-1) + A[j]\}$
end for

6.3.4

- You are trying to decide where to build a chain of restaurants along a linear highway.
- At each location i along the highway, you will make a profit of $p(i)$ that depends on the location.
- You are not allowed to put two restaurants within k miles of each other.
- Where should you place the restaurants to maximize profit?
- Hint: Draw the DAG!

My Answer



$dist(j)$ distance vector of j
$P[j]$ max profit ending at index j
$P[j] = max\{p(i), p(j-1) + A[j]\}$
$P[0] = 0$
$dist(0) = 0$
for all $j = 1, 2, ...n$, in linearized order **do**
    $P[j] = max\{p(i), p(j-1) + A[j]\}$
    $dist(j) = min\{(j, j-1)_k, dist(j)\}$
end for