Dijkstra's algorithm

CS 312

Dijkstra's algorithm

\[ L[i,j] = q : \text{the weight along path from } i \text{ to } j \text{ is } q. \]
\[ D[k] = m : \text{length of the shortest special path from source to } k. \]
\[ j \text{ in } C : \text{shortest path from source to } j \text{ is not known.} \]
\[ k \text{ in } S : \text{shortest path from source to } k \text{ is known.} \]

Dijkstra \((L[1 \ldots n, 1 \ldots n])\): array \([2 \ldots n]\)
array \(D[2 \ldots n]\)
\(C \leftarrow (2, 3 \ldots n)\)
for \(i \leftarrow 2 \text{ to } n \) do \(D[i] \leftarrow L[1,i] \)
repeat \(n \cdot 2 \) times
\(v \leftarrow \text{some element of } C \text{ minimizing } D[v]\)
\(C \leftarrow C \setminus (v)\)
for each \(w \text{ in } C \) do
\(D[w] \leftarrow \min (D[w], D[v] + L[v,w])\)
return \(D\)

Knapsack problem I

Given \(n\) objects.
Each object has a value, \(v_i\), and a weight, \(w_i\).
Maximize the value of the stuff in the knapsack.

\[ \text{maximize } \sum_{i=1}^{n} v_i x_i \text{ but keep } \sum_{i=1}^{n} x_i w_i \leq W \]

Knapsack

\[ \text{function } \text{knapsack} (w[1 \ldots n], v[1 \ldots n], W) : \text{array } [1 \ldots n] \]
for \(i \leftarrow 1 \text{ to } n \) do \(x[i] \leftarrow 0\)
weight \(\leftarrow 0\)
while weight \(<\ W \) do
\(i \leftarrow \text{the best remaining object}\)
if weight + \(w[i] \leq W \) then \(x[i] \leftarrow 1\)
\(\text{weight } \leftarrow \text{weight } + w[i]\)
else \(x[i] \leftarrow (W - \text{weight})/w[i]\)
\(\text{weight } \leftarrow W\)
return \(x\)