CS 577- Intro to Algorithms

Dynamic Programming (Part 4)

Dieter van Melkebeek

October 1, 2020

Outline

Outline

Paradigm

Recursive approach such that:

1. The number of distinct subproblems in the recursion tree is small.
2. Each of those subproblems is solved only once.

Outline

Paradigm

Recursive approach such that:

1. The number of distinct subproblems in the recursion tree is small.
2. Each of those subproblems is solved only once.

Examples

- Computing Fibonacci numbers
- Weighted interval scheduling
- Knapsack problem
- RNA secondary structure
- Sequence alignment / Edit distance

Outline

Paradigm

Recursive approach such that:

1. The number of distinct subproblems in the recursion tree is small.
2. Each of those subproblems is solved only once.

Examples

- Computing Fibonacci numbers
- Weighted interval scheduling
- Knapsack problem
- RNA secondary structure
- Sequence alignment / Edit distance
- Shortest paths

Shortest Paths Problem

Shortest Paths Problem

Input: (di)graph $G = (V, E)$; lengths $\ell : E \to \mathbb{R}$; $s, t \in V$

Shortest Paths Problem

Input: (di)graph $G = (V, E)$; lengths $\ell : E \to \mathbb{R}$; $s, t \in V$

Ouput: path P from s to t with minimum length $\ell(P) \doteq \sum_{e \in P} \ell(e)$

Shortest Paths Problem

Input: (di)graph $G = (V, E)$; lengths $\ell : E \to \mathbb{R}$; $s, t \in V$

Ouput: path P from s to t with minimum length $\ell(P) \doteq \sum_{e \in P} \ell(e)$

Variants based on source/target

Shortest Paths Problem

Input: (di)graph $G = (V, E)$; lengths $\ell : E \to \mathbb{R}$; $s, t \in V$

Ouput: path P from s to t with minimum length $\ell(P) \doteq \sum_{e \in P} \ell(e)$

Variants based on source/target

- single pair

Shortest Paths Problem

Input: (di)graph $G = (V, E)$; lengths $\ell : E \to \mathbb{R}$; $s, t \in V$

Ouput: path P from s to t with minimum length $\ell(P) \doteq \sum_{e \in P} \ell(e)$

Variants based on source/target

- single pair
- single source

Shortest Paths Problem

Input: (di)graph $G = (V, E)$; lengths $\ell : E \to \mathbb{R}$; $s, t \in V$

Ouput: path P from s to t with minimum length $\ell(P) \doteq \sum_{e \in P} \ell(e)$

Variants based on source/target

- single pair
- single source
- single target

Shortest Paths Problem

Input: (di)graph $G = (V, E)$; lengths $\ell : E \to \mathbb{R}$; $s, t \in V$

Ouput: path P from s to t with minimum length $\ell(P) \doteq \sum_{e \in P} \ell(e)$

Variants based on source/target

- single pair
- single source
- single target
- all pairs

Shortest Paths Problem

Input: (di)graph $G = (V, E)$; lengths $\ell : E \to \mathbb{R}$; $s, t \in V$

Ouput: path P from s to t with minimum length $\ell(P) \doteq \sum_{e \in P} \ell(e)$

Variants based on source/target

- single pair
- single source
- single target
- all pairs

Variants based on edge lengths

Shortest Paths Problem

Input: (di)graph $G = (V, E)$; lengths $\ell : E \to \mathbb{R}$; $s, t \in V$

Ouput: path P from s to t with minimum length $\ell(P) \doteq \sum_{e \in P} \ell(e)$

Variants based on source/target

- single pair
- single source
- single target
- all pairs

Variants based on edge lengths

- unit

Shortest Paths Problem

Input: (di)graph $G = (V, E)$; lengths $\ell : E \to \mathbb{R}$; $s, t \in V$

Ouput: path P from s to t with minimum length $\ell(P) \doteq \sum_{e \in P} \ell(e)$

Variants based on source/target

- single pair
- single source
- single target
- all pairs

Variants based on edge lengths

- unit
- nonnegative

Shortest Paths Problem

Input: (di)graph $G = (V, E)$; lengths $\ell : E \to \mathbb{R}$; $s, t \in V$

Ouput: path P from s to t with minimum length $\ell(P) \doteq \sum_{e \in P} \ell(e)$

Variants based on source/target

- single pair
- single source
- single target
- all pairs

Variants based on edge lengths

- unit
- nonnegative
- arbitrary

Shortest Paths Problem

Specification

Input: (di)graph $G = (V, E)$; lengths $\ell : E \to \mathbb{R}$; $s, t \in V$

Ouput: path P from s to t with minimum length $\ell(P) \doteq \sum_{e \in P} \ell(e)$

Shortest Paths Problem

Specification

Input: (di)graph $G = (V, E)$; lengths $\ell : E \to \mathbb{R}$; $s, t \in V$

Ouput: path P from s to t with minimum length $\ell(P) \doteq \sum_{e \in P} \ell(e)$

Distance $d(s, t)$

Shortest Paths Problem

Specification

Input: (di)graph $G = (V, E)$; lengths $\ell : E \to \mathbb{R}$; $s, t \in V$

Ouput: path P from s to t with minimum length $\ell(P) \doteq \sum_{e \in P} \ell(e)$

Distance $d(s, t)$

$= \min\{\ell(P) \mid P \text{ path from } s \text{ to } t\}$

Shortest Paths Problem

Specification

Input: (di)graph $G = (V, E)$; lengths $\ell : E \to \mathbb{R}$; $s, t \in V$

Ouput: path P from s to t with minimum length $\ell(P) \doteq \sum_{e \in P} \ell(e)$

Distance $d(s, t)$

$= \min\{\ell(P) \mid P \text{ path from } s \text{ to } t\}$

$= \infty$ if there is no path from s to t

Shortest Paths Problem

Specification

Input: (di)graph $G = (V, E)$; lengths $\ell : E \to \mathbb{R}$; $s, t \in V$

Ouput: path P from s to t with minimum length $\ell(P) \doteq \sum_{e \in P} \ell(e)$

Distance $d(s, t)$

$= \min\{\ell(P) \mid P \text{ path from } s \text{ to } t\}$

$= \infty$ if there is no path from s to t

$= -\infty$ if there is a path from s to t but no shortest one

Shortest Paths Problem

Specification

Input: (di)graph $G = (V, E)$; lengths $\ell : E \to \mathbb{R}$; $s, t \in V$

Ouput: path P from s to t with minimum length $\ell(P) \doteq \sum_{e \in P} \ell(e)$

Distance $d(s,t)$

$= \min\{\ell(P) \mid P$ path from s to $t\}$
$= \infty$ if there is no path from s to t
$= -\infty$ if there is a path from s to t but no shortest one

Proposition

$d(s,t) = -\infty \Leftrightarrow$
there exists cycle C with $\ell(C) < 0$ such that $s \rightsquigarrow C$ and $C \rightsquigarrow t$.

Principle of optimality

Single Source – subproblems and recurrence

Subproblems

$\mathrm{OPT}(k, v) =$

length of a shortest path from s to v using $\leq k$ edges

∞ if no such path exists

Single Source – subproblems and recurrence

Subproblems

$\mathrm{OPT}(k, v) =$

length of a shortest path from s to v using $\leq k$ edges

∞ if no such path exists

$(k \in \mathbb{N}$ and $v \in V)$

Single Source – subproblems and recurrence

Subproblems

$\mathrm{OPT}(k, v) =$

length of a shortest path from s to v using $\leq k$ edges

∞ if no such path exists

$(k \in \mathbb{N}$ and $v \in V)$

Base case $(k = 0)$

Single Source – subproblems and recurrence

Subproblems

$\mathrm{OPT}(k, v) =$

length of a shortest path from s to v using $\leq k$ edges

∞ if no such path exists

($k \in \mathbb{N}$ and $v \in V$)

Base case ($k = 0$)

$\mathrm{OPT}(0, s) = 0$ and $\mathrm{OPT}(0, v) = \infty$ for $v \neq s$

Single Source – subproblems and recurrence

Subproblems

$\mathrm{OPT}(k, v) =$

length of a shortest path from s to v using $\leq k$ edges

∞ if no such path exists

$(k \in \mathbb{N}$ and $v \in V)$

Base case $(k = 0)$

$\mathrm{OPT}(0, s) = 0$ and $\mathrm{OPT}(0, v) = \infty$ for $v \neq s$

Recursive case $(k \geq 1)$

Single Source – subproblems and recurrence

Subproblems

$\mathrm{OPT}(k, v) =$
  length of a shortest path from s to v using $\leq k$ edges
  ∞ if no such path exists
$(k \in \mathbb{N}$ and $v \in V)$

Base case $(k = 0)$

$\mathrm{OPT}(0, s) = 0$ and $\mathrm{OPT}(0, v) = \infty$ for $v \neq s$

Recursive case $(k \geq 1)$

$\mathrm{OPT}(k, v) =$
  $\min_{(u,v) \in E}(\mathrm{OPT}(k-1, u) + \ell(u, v))$

Single Source – subproblems and recurrence

Subproblems

$\mathrm{OPT}(k, v) =$

length of a shortest path from s to v using $\leq k$ edges

∞ if no such path exists

$(k \in \mathbb{N}$ and $v \in V)$

Base case $(k = 0)$

$\mathrm{OPT}(0, s) = 0$ and $\mathrm{OPT}(0, v) = \infty$ for $v \neq s$

Recursive case $(k \geq 1)$

$\mathrm{OPT}(k, v) =$

$\min_{(u,v)\in E}(\mathrm{OPT}(k-1, u) + \ell(u, v))$

$\mathrm{OPT}(k-1, v)$

Single Source – subproblems and recurrence

Subproblems

$\mathrm{OPT}(k, v) =$

length of a shortest path from s to v using $\leq k$ edges

∞ if no such path exists

$(k \in \mathbb{N}$ and $v \in V)$

Base case $(k = 0)$

$\mathrm{OPT}(0, s) = 0$ and $\mathrm{OPT}(0, v) = \infty$ for $v \neq s$

Recursive case $(k \geq 1)$

$\mathrm{OPT}(k, v) =$

$\min_{(u,v) \in E}(\mathrm{OPT}(k-1, u) + \ell(u, v))$

$\mathrm{OPT}(k-1, v)$

Answer

$d(s, v)$

Single Source – subproblems and recurrence

Subproblems

$\mathrm{OPT}(k, v) =$
  length of a shortest path from s to v using $\leq k$ edges
  ∞ if no such path exists
($k \in \mathbb{N}$ and $v \in V$)

Base case ($k = 0$)

$\mathrm{OPT}(0, s) = 0$ and $\mathrm{OPT}(0, v) = \infty$ for $v \neq s$

Recursive case ($k \geq 1$)

$\mathrm{OPT}(k, v) =$
  $\min_{(u,v) \in E}(\mathrm{OPT}(k-1, u) + \ell(u, v))$
  $\mathrm{OPT}(k-1, v)$

Answer

$d(s, v) = \lim_{k \to \infty} \mathrm{OPT}(k, v)$

Single Source – number of iterations

Single Source – number of iterations

Observation 1
If $(\forall v)\, \mathrm{OPT}(k, v) = \mathrm{OPT}(k-1, v)$
then $(\forall v)\, \mathrm{OPT}(k+1, v) = \mathrm{OPT}(k, v)$.

Single Source – number of iterations

Observation 1

If $(\forall v)\, \mathrm{OPT}(k, v) = \mathrm{OPT}(k-1, v)$
then $(\forall v)\, \mathrm{OPT}(k+1, v) = \mathrm{OPT}(k, v)$.

Note

If $\ell \geq 0$, the hypothesis of Observation 1 holds for $k = n$.

Single Source – number of Iterations

Observation 2

If $\mathrm{OPT}(n, v) < \mathrm{OPT}(n-1, v)$ then
there exists cycle C with $\ell(C) < 0$ such that $s \rightsquigarrow C$ and $C \rightsquigarrow v$.

Single Source – number of Iterations

Observation 2

If $\text{OPT}(n, v) < \text{OPT}(n-1, v)$ then
there exists cycle C with $\ell(C) < 0$ such that $s \rightsquigarrow C$ and $C \rightsquigarrow v$.

Proof

$$\begin{aligned}
&\text{OPT}(n, v) \\
&\quad = \ell(P) \\
&\quad = \ell(P - C) + \ell(C) \\
&\quad \geq \text{OPT}(n-1, v) + \ell(C) \\
&\quad > \text{OPT}(n, v) + \ell(C) \\
&\Rightarrow \ell(C) < 0
\end{aligned}$$

Single Source – number of Iterations

Observation 1

If $(\forall v)\, \mathrm{OPT}(k, v) = \mathrm{OPT}(k-1, v)$
then $(\forall v)\, \mathrm{OPT}(k+1, v) = \mathrm{OPT}(k, v)$.

Observation 2

If $\mathrm{OPT}(n, u) < \mathrm{OPT}(n-1, u)$ then
there exists cycle C with $\ell(C) < 0$ such that $s \rightsquigarrow C$ and $C \rightsquigarrow u$.

Criterion

$d(s, v) = -\infty \Leftrightarrow$
there exist $u \in V$ s.t. $u \rightsquigarrow v$ and $\mathrm{OPT}(n, u) < \mathrm{OPT}(n-1, u)$.

Single Source – algorithm

Single Source – algorithm

Recurrence

- $\text{OPT}(0, s) = 0$ and $\text{OPT}(0, v) = \infty$ for $v \neq s$
- $\text{OPT}(k, v) =$
  $\min\big(\text{OPT}(k-1, v), \min_{(u,v)\in E}(\text{OPT}(k-1, u) + \ell(u, v))\big)$

Single Source – complexity analysis

Recurrence

- $\mathrm{OPT}(0, s) = 0$ and $\mathrm{OPT}(0, v) = \infty$ for $v \neq s$
- $\mathrm{OPT}(k, v) =$
  $\min\left(\mathrm{OPT}(k-1, v), \min_{(u,v)\in E}(\mathrm{OPT}(k-1, u) + \ell(u, v))\right)$

Time

- $O(n+m)$ per row
- $\leq n+1$ rows
- $O(n(n+m))$ total

Space

- $O(n)$ for computing distances
- $O(n^2) \to O(n)$ for finding shortest paths

Principle of optimality

Principle of optimality

$V = [n]$

All Pairs – subproblems

Subproblems

$\mathrm{OPT}(s, t, k) =$
length of a shortest path from s to t
that only uses $\{k, \ldots, n\}$ as intermediate vertices

All Pairs – subproblems

Subproblems

$\text{OPT}(s, t, k) =$

length of a shortest path from s to t

that only uses $\{k, \ldots, n\}$ as intermediate vertices

∞ if no such path exists

All Pairs – subproblems

Subproblems

$\mathrm{OPT}(s, t, k) =$

length of a shortest path from s to t
that only uses $\{k, \dots, n\}$ as intermediate vertices

∞ if no such path exists

$-\infty$ if there is such a path but no shortest one

All Pairs – subproblems

Subproblems

$\text{OPT}(s, t, k) =$

length of a shortest path from s to t

that only uses $\{k, \ldots, n\}$ as intermediate vertices

∞ if no such path exists

$-\infty$ if there is such a path but no shortest one

$(s, t \in V$ and $k \in [n + 1])$

All Pairs – subproblems

Subproblems

$\mathrm{OPT}(s, t, k) =$

length of a shortest path from s to t

that only uses $\{k, \ldots, n\}$ as intermediate vertices

∞ if no such path exists

$-\infty$ if there is such a path but no shortest one

$(s, t \in V$ and $k \in [n+1])$

Base case $(k = n + 1)$

All Pairs – subproblems

Subproblems

$\mathrm{OPT}(s, t, k) =$
length of a shortest path from s to t
that only uses $\{k, \ldots, n\}$ as intermediate vertices
∞ if no such path exists
$-\infty$ if there is such a path but no shortest one
$(s, t \in V$ and $k \in [n+1])$

Base case $(k = n + 1)$

- $\mathrm{OPT}(s, t, n+1) = 0$ if $s = t$

All Pairs – subproblems

Subproblems

$\mathrm{OPT}(s, t, k) =$

length of a shortest path from s to t

that only uses $\{k, \ldots, n\}$ as intermediate vertices

∞ if no such path exists

$-\infty$ if there is such a path but no shortest one

$(s, t \in V$ and $k \in [n+1])$

Base case $(k = n + 1)$

- $\mathrm{OPT}(s, t, n+1) = 0$ if $s = t$
- $\mathrm{OPT}(s, t, n+1) = \ell(s, t)$ if $(s, t) \in E$

All Pairs – subproblems

Subproblems

$\mathrm{OPT}(s, t, k) =$

length of a shortest path from s to t

that only uses $\{k, \ldots, n\}$ as intermediate vertices

∞ if no such path exists

$-\infty$ if there is such a path but no shortest one

($s, t \in V$ and $k \in [n+1]$)

Base case ($k = n + 1$)

- $\mathrm{OPT}(s, t, n+1) = 0$ if $s = t$
- $\mathrm{OPT}(s, t, n+1) = \ell(s, t)$ if $(s, t) \in E$
- $\mathrm{OPT}(s, t, n+1) = \infty$ otherwise

All Pairs – subproblems

Subproblems

$\mathrm{OPT}(s, t, k) =$

length of a shortest path from s to t

that only uses $\{k, \dots, n\}$ as intermediate vertices

∞ if no such path exists

$-\infty$ if there is such a path but no shortest one

($s, t \in V$ and $k \in [n+1]$)

Base case ($k = n + 1$)

- $\mathrm{OPT}(s, t, n+1) = 0$ if $s = t$
- $\mathrm{OPT}(s, t, n+1) = \ell(s, t)$ if $(s, t) \in E$
- $\mathrm{OPT}(s, t, n+1) = \infty$ otherwise

Answer

$d(s, t) = \mathrm{OPT}(s, t, 1)$

All Pairs – recurrence and analysis

All Pairs – recurrence and analysis

Recursive case ($k \leq n$)

All Pairs – recurrence and analysis

Recursive case ($k \leq n$)

If $\text{OPT}(s, k, k+1) < \infty$ and $\text{OPT}(k, t, k+1) < \infty$
and $\text{OPT}(k, k, k+1) < 0$
then $\text{OPT}(s, t, k) = -\infty$
else $\text{OPT}(s, t, k) =$
$\min(\text{OPT}(s, t, k+1), \text{OPT}(s, k, k+1) + \text{OPT}(k, t, k+1))$

All Pairs – recurrence and analysis

Recursive case $(k \leq n)$

If $\mathrm{OPT}(s, k, k+1) < \infty$ and $\mathrm{OPT}(k, t, k+1) < \infty$
  and $\mathrm{OPT}(k, k, k+1) < 0$
then $\mathrm{OPT}(s, t, k) = -\infty$
else $\mathrm{OPT}(s, t, k) =$
  $\min(\mathrm{OPT}(s, t, k+1), \mathrm{OPT}(s, k, k+1) + \mathrm{OPT}(k, t, k+1))$

Time

- $O(1)$ per entry
- $O(n^3)$ entries
- $O(n^3)$ total

All Pairs – recurrence and analysis

Recursive case ($k \le n$)

If $\mathrm{OPT}(s, k, k+1) < \infty$ and $\mathrm{OPT}(k, t, k+1) < \infty$
and $\mathrm{OPT}(k, k, k+1) < 0$
then $\mathrm{OPT}(s, t, k) = -\infty$
else $\mathrm{OPT}(s, t, k) =$
$\min(\mathrm{OPT}(s, t, k+1), \mathrm{OPT}(s, k, k+1) + \mathrm{OPT}(k, t, k+1))$

Time

- $O(1)$ per entry
- $O(n^3)$ entries
- $O(n^3)$ total

Space

$O(n^2)$ for distances and shortest paths