RWTH Aachen University **WS 2010/11**
Lehrgebiet Theoretische Informatik Tutorial 6
Kneis—Langer—Sikdar 23.11.2010

Tutorial Exact Algorithms

Exercise T12 TSP in Bounded Degree Graphs. In class, we saw that n-vertex graphs with maximum degree Δ have at most

$$[(2^{\Delta+1}-1)^{1/(\Delta+1)}]^n$$

connected vertex sets. This is better than the trivial 2^n bound for general graphs: for $\Delta = 3$, for instance, this bound works out to 1.97^n. However the algorithm requires a data structure that stores connected set families of size i, where $1 \leq i \leq n$, that supports the following operations:

- insert a new connected set in polynomial time (polynomial in n, where n is the number of vertices);
- output all connected sets of a particular size in time proportional to their total size;
- given a connected set S, we should be able to read off $\mathrm{OPT}[S; c_i]$ for all $c_i \in S$ in time $O(|S|)$.

In addition, the size of the data structure should be proportional to the total size of all the connected sets. Design an appropriate data structure for this algorithm.

Exercise T13 Minimum Exact Set Cover. This problem is defined as follows: given a finite set $\mathcal{U} = \{1, \ldots, n\}$ and a collection $\mathcal{F}$ of $m > 0$ subsets of $\mathcal{U}$, you have to find a subset $\mathcal{F}' \subseteq \mathcal{F}$ of minimum size that partitions $\mathcal{U}$. That is, the sets in $\mathcal{F}'$ must be pairwise disjoint and their union must be the whole set $\mathcal{U}$. Design a DP algorithm for this problem with running time $O(2^n \cdot nm)$.

Solution For $X \subseteq \mathcal{U}$, define $\mathrm{OPT}[X]$ to be the size of a smallest subset of $\mathcal{F}$ that partitions X. Then $\mathrm{OPT}[\mathcal{U}]$ represents the size of the optimal solution.

- $\mathrm{OPT}[\emptyset] = 0$;
- For $X \neq \emptyset$, fix an arbitrary element $i \in X$. We then have

$$\mathrm{OPT}[X] = \min_{S \in \mathcal{F}, i \in S} \{\mathrm{OPT}[X \setminus S] + 1\}.$$

The time taken to compute $\mathrm{OPT}[X]$ is $O(mn)$ and the total time taken to compute the optimal solution is then $O(2^n \cdot mn)$.

Exercise T14 Longest Common Subsequence. Let Σ be a finite alphabet and let $\mathbf{x} = x_1 x_2 \dots x_n$ and $\mathbf{y} = y_1 y_2 \dots y_m$ be strings from Σ^*. A subsequence of $\mathbf{x}$ is a string of characters $x_{i_1} x_{i_2} \dots x_{i_p}$ from the sequence $\mathbf{x}$ such that $i_1 < i_2 < \cdots < i_p$. For example, if $\mathbf{x} =$ `geranium` then `germ` is a subsequence of $\mathbf{x}$. A sequence that is a subsequence of both $\mathbf{x}$ and $\mathbf{y}$ is a *common subsequence*. A common subsequence of maximum length is a *longest common subsequence* of $\mathbf{x}$ and $\mathbf{y}$. Design a DP algorithm to find out the longest common subsequence of $\mathbf{x}$ and $\mathbf{y}$ in time $O(mn)$.

Solution For $1 \le i \le n$ and $1 \le j \le m$, define $\mathrm{OPT}[i, j]$ to be the length of the longest common subsequence of $x_1 \dots x_i$ and $y_1 \dots y_j$. Then

$$\mathrm{OPT}[i,j] = \begin{cases} \mathrm{OPT}[i-1, j-1] + 1 & \text{if } x_i = y_i; \\ \max\{\mathrm{OPT}[i-1, j], \mathrm{OPT}[i, j-1]\}, & \text{otherwise.} \end{cases}$$

The time taken to compute $\mathrm{OPT}[n, m]$ is $O(nm)$.

Homework Assignment H13 (10 Points) Consider the Exact SAT problem: given a Boolean formula in n variables and m clauses in CNF, find a satisfying assignment, if any, such that in every clause exactly one literal is true.

1. Give a reduction from Exact SAT to Min Exact Hitting Set which is defined as follows. Given a collection $\mathcal{F}$ of subsets of a finite universe $\mathcal{U}$, find a subset $X \subseteq \mathcal{U}$ of minimum size such that every set in $\mathcal{F}$ contains exactly one element of X.
2. Show that the Min Exact Hitting Set problem can be solved in time $O^*(2^m)$, where $m = |\mathcal{F}|$.

This will show that Exact SAT can be solved in time $O^*(2^m)$ time.

Solution

1. Given an instance ϕ of Exact SAT with n variables and m clauses, define an instance $(\mathcal{U}, \mathcal{F})$ of Min Exact Hitting Set as follows. The universe $\mathcal{U}$ is the set of all literals (positive and negative occurences of variables) that occur in ϕ; the set $\mathcal{F}$ is the union of the set of clauses of ϕ and

   $$\{\{x, \bar{x}\} : x \text{ or } \bar{x} \text{ occurs in } \phi\}.$$

   If there exists an assignment to the variables of ϕ that sets exactly one literal of each clause to true, then this set of literals obviously contains exactly one literal from each set of $\mathcal{F}$. Conversely, given a set $X \subseteq \mathcal{U}$ of elements such that $|X \cap S| = 1$ for each $S \in \mathcal{F}$, it follows that it cannot contain both the negative and positive occurrences of any variable. It is easy to see that the assignment that sets the literals of X to 1 and the rest to 0 is a satisfying assignment for ϕ with the property that exactly one literal per clause is set to 1.

Homework Assignment H14 (10 Points) Longest Common Subsequence of 3 Strings. Given three strings $\mathbf{x}, \mathbf{y}, \mathbf{z}$ over a finite alphabet Σ, design an algorithm that finds out the longest common subsequence of these strings. What is the running time of your algorithm?

Solution Let $\mathbf{x} = x_1, \ldots, x_p$, $\mathbf{y} = y_1, \ldots, y_q$, and $\mathbf{z} = z_1, \ldots, z_r$. For $1 \leq i \leq p$, $1 \leq j \leq q$, and $1 \leq k \leq r$, define $\mathrm{OPT}[i, j, k]$ to be the length of the maximum subsequence of the strings $\mathbf{x}[1, \ldots, i]$, $\mathbf{y}[1, \ldots, j]$, and $\mathbf{z}[1, \ldots, k]$. One can easily compute $\mathrm{OPT}[1, 1, 1]$, $\mathrm{OPT}[i, 1, 1]$, $\mathrm{OPT}[1, j, 1]$, $\mathrm{OPT}[1, 1, k]$. Now for computing $\mathrm{OPT}[i, j, k]$ for $i, j, k \geq 2$ there are several cases to consider.

1. $x_i = y_j = z_k$: In this case, the last element of each of the sequences is part of the longest common subsequence. Hence

$$\mathrm{OPT}[i, j, k] = \mathrm{OPT}[i - 1, j - 1, k - 1] + 1.$$

2. $x_i = y_j \neq z_k$: In this case, either x_i is part of the longest subsequence or not and we have

$$\mathrm{OPT}[i, j, k] = \max\{\mathrm{OPT}[i - 1, j - 1, k], \mathrm{OPT}[i, j, k - 1]\}.$$

   There are two more cases corresponding to the conditions $x_i = z_k \neq y_j$ and $x_i \neq y_j = z_k$.

3. Finally consider the case $x_i \neq y_j \neq z_k$. In this situation, either x_i or y_j or z_k or none of them is the last symbol of the subsequence. Hence $\mathrm{OPT}[i, j, k]$ is the maximum of the quantities $\mathrm{OPT}[i, j - 1, k - 1]$, $\mathrm{OPT}[i - 1, j, k - 1]$, $\mathrm{OPT}[i - 1, j - 1, k]$, and $\mathrm{OPT}[i - 1, j - 1, k - 1]\}$.

Computing $\mathrm{OPT}[i, j, k]$ for all i, j, k takes time $O(pqr)$.