15-251: Great Theoretical Ideas In Computer Science

Recitation FINAL Solutions

Reductions:

- **Polynomial Time** We say that an algorithm runs in **Polynomial Time** if, for some constant $c$, its running time is $O(n^c)$, where $n$ is the size of the input.

- **Polytime Reducible** We say that problem $A$ is **polytime reducible** to problem $B$ (we write $A \leq_p B$) if we can solve problem $A$ in polynomial time given a polynomial time black-box algorithm for problem $B$. Problem $A$ is **polytime equivalent** to problem $B$ ($A =_p B$) if $A \leq_p B$ and $B \leq_p A$.

- **P** We define **P** as the set of decision problems solvable in polynomial time. Examples include: 2-COLOR (Given a graph $G$, does there exist a 2-coloring of $G$?), BIPARTITE-MATCHING (Given a bipartite graph $G = (X, Y, E)$, does there exist a perfect matching for $G$).

- **NP** We define **NP** as the set of decision problems that have polynomial-time verifiers. That is, if a problem is in **NP**, then there exists a polynomial time algorithm $V$ such that if $I$ is a YES-instance, then there exists $X$ such that $V(I, X) = YES$ and if $I$ is a NO-instance, then for all $X$, $V(I, X) = NO$.

- **NP-hard** A problem $Q$ is **NP-hard** if for any other problem $Q'$ in **NP**, $Q' \leq_p Q$.

- **NP-complete** A problem $Q$ is **NP-complete** if $Q$ is in **NP** and $Q$ is **NP-hard**. **NP-complete** problems that you should know: SAT, 3-SAT, 3-COLOR, HAM, CLIQUE, INDEPENDENT-SET, SUDOKU.

- Note that the set **P** and **NP** consist of decision problems. That is, the answer to the problem is always either **YES** or **NO**. Therefore, search problems are neither in **P** nor in **NP**. Example: “What is the maximum $k$ such that $G$ has a $k$-clique” is **NOT** in **P** neither is it in **NP**.

- **P $\subseteq$ NP**. But no one knows if **NP $\subseteq$ P**.

- Recipe to prove that a problem $Q$ is **NP-complete**:
  
  (i) First prove that you can verify proofs for $Q$ in polynomial time.
  
  (ii) Assume you have an ORACLE to solve $Q$ in polynomial time.
  
  (iii) Choose your favorite **NP-complete** problem $C$ (Hint: Don’t try to use SUDOKU).
  
  (iv) In polynomial time, transform an instance $X$ of $C$ to an instance $X'$ of $Q$.
  
  (v) The transformation must be such that if $X'$ is a YES-instance of $Q$ then $X$ is a YES-instance of $C$, and if $X'$ is a NO-instance of $Q$ then $X$ is a NO-instance of $C$.
  
  (vi) Therefore, since we had a polynomial ORACLE for $Q$ we can solve $C$ in polynomial time. [The trick is usually choosing the right $C$, sometimes it might not be your favorite one :()
Candy-Cane-Cover-Consistency

VERTEX COVER is the following problem: given a graph $G = (V, E)$ and an integer $k$, does there exist a set $S = \{ v : v \in V \}$ such that $\forall e \in E \exists v \in V : v \in e$ and $|S| \leq k$.

(i) Show that VERTEX COVER is in $\text{NP}$

(ii) Show that VERTEX COVER is $\text{NP}$-hard by reducing from INDEPENDENT-SET (Just ask $\text{VC}(G, n-k)$)

(iii) Show that CCC (Candy-Cane-Consistency) is $\text{NP}$-hard by reducing from VERTEX COVER (Hint: use the same idea from the reference solution)

I Love Transistors

Define sets $A, B, C$ such that $A \subseteq B \subseteq C$. If it is $\text{NP}$-complete to determine membership of $A$ and $C$, must it be $\text{NP}$-complete to determine membership of $B$? (Hint: You may use the fact that 1-in-3 SAT is $\text{NP}$-complete, that is given an instance of 3-SAT, is there an assignment of variables such that only one literal per clause is satisfied? Another Hint: solving systems of linear equations is in $\text{P}$)

Sketch: Let $B$ be the set of CNF formulas that have an assignment of variables such that an odd number of literals are satisfied. Use addition modulo 2 and solve the system of linear equations.

Generating Functions:

- Basically represent a sequence $< a_0, a_1, \ldots >$ as a formal power series: $\sum_n a_n x^n$. Given a formal power series, the coefficient of $x^n$ is $a_n$.
- $\sum_{n=0}^{\infty} x^n = \frac{1}{1-x}$.
- General strategy for BT, MT GF questions:
  
  (i) Find a recurrence for the problem. Do a counting argument to justify it (also work small examples to make sure it works!)
  
  (ii) $A(x) = \sum_n a_n x^n = \text{Base case(s)} + \sum \text{Recurrence} \cdot x^n$
  
  (iii) Do a lot of algebra to find $A(x)$ in terms of $x$.
  
  (iv) If necessary, use partial fractions to expand $A(x)$. Find its formal power series and find the coefficient of $x^n$ for $a_n$.
  
  (v) Don’t forget that sometimes the answer might be sneaky! (Differentiation).

You Thought You Were Done With 251

The 251 TAs finally gained control of all the Robot Unicorns, Nyan Cats and Double Rainbows in the universe! Since they got bored of writing 251 exams, they decided to start arranging their Robot Unicorns, Nyan Cats and Double Rainbows in a line. Of course, Nyan Cats and Robot Unicorns don’t
get along well together. Help the TAs figure out how many ways there are to make a line of length \( n \) such that there is no adjacent pair containing a Robot Unicorn and a Nyan Cat. (Hint: case on the end of the line)

| Sketch: The recurrence is \( T_n = 2T_{n-1} + T_{n-2} \). We argued by making three different recurrences for R,N,D and then observing that \( T = R + N + D \). |

### Graphs:

- A graph \( G \) is a tuple \((V, E)\) such that \( V \) is the set of vertices of the graph and \( E \) is the set of edges of the graph, where \( e \in E \iff e = (u, v) \) for some \( u, v \in V \). In this class we only consider simple graphs (no self-loops or multi-edges).
- Review the multiple definition of trees. One of them is \( G \) is connected and \( n = e + 1 \).
- Cayley’s formula: the number of labeled trees on \( n \) nodes is \( n^{n-2} \). Review Prüfer codes.
- Spanning trees: a tree \( T \) is a spanning tree of a graph \( G \) if \( V(T) = V(G) \) and \( E(T) \cup E(G) = E(T) \).
- Planar graphs: are graphs isomorphic to a plane graph. In other words, planar graphs are graphs that can be drawn in a plane without crossing edges.
- Euler’s formula: \( n - e + f = 2 \) for connected, planar graphs.
- Review Kruskal’s algorithm.
- A graph \( G \) is bipartite if there exists a bipartition \( V = (X, Y) \) of the vertex set such that there is no edge \( e = (u, v) \) such that \( u, v \in X \) or \( u, v \in Y \), and \( |X|, |Y| > 0 \)
- Perfect Matching: A matching \( M \) is a set of edges, no two which share a vertex. A matching is perfect if it includes every edge.
- Hall’s Marriage Theorem states that a bipartite graph has a perfect matching if and only if \( |X| = |Y| \) and \( \forall S \subseteq X, |N(S)| \geq |S| \) (Actually Hall’s Theorem is more general, you only need to know this special case).

### But Cayley’s Dead

Given a vertex set \( v_1, v_2, \ldots, v_n \). How many labeled trees on \( n \) vertices include the edge \((v_1, v_2)\)? (Hint: consider the title)

| Sketch: \( \binom{V}{2} x = (n - 1)n^{n-2} \). Count the number of edges in all possible trees and distribute them among the edges. |

### Drawing Circles

Given a connected graph on \( n \) vertices where each vertex is of degree at least \( d \) and \( d < \frac{n}{2} \), there exists a simple path of length at least \( 2d \) (hence, at least \( 2d + 1 \) distinct vertices are on this path). (Hint: use a similar idea as the last graph problem on Test 3).
Sketch: Take a maximal path and show that there's a cycle. Since the graph is connected it is possible to arrive from some node not on the cycle to the cycle and then traversing the cycle, thus making a longer path.

Lohengrin

Let $S_1, S_2, \ldots, S_m$ and $T_1, T_2, \ldots, T_m$ be two partitions of the set $X$ into sets of size $k$. Show that there exists $s_1, s_2, \ldots, s_m$ that is a set of distinct representative for both partitions.

Sketch: Construct 2 sets of vertices containing all elements of $X$. The bipartite (multi)-graph has same degree for each vertex and invoke Hall’s Marriage Theorem.

But We Never Learned Ramsey

Given $K_n$ such that $n = s + t$. Let $P$ be a random red-blue coloring of $K_n$. Show that either there is a vertex incident to $s$ red edges or there is a blue path of length $t$.

Sketch: Assume that no vertex is incident to $\geq s$ red edges then every vertex has blue degree $\geq t$. Then there is a blue path of length $t$.

Probability:

- **Sample Space** The set of all outcomes, usually denoted by $S$ or $\Omega$.
- **Event** A set $E \subseteq \Omega$.
- **Uniform Distribution** If each element has equal probability, then $P[E] = \frac{|E|}{|\Omega|}$.
- **Review axioms of probability.**
- **Conditional Probability** $P[A|B] = \frac{P[A \cap B]}{P[B]}$.
- **Disjoint** $P[A \cap B] = 0$.
- **Law of Total Probability** $P[A] = \sum_{i=1}^{n} P[A \cap E_i]P[E_i]$
- **Bayes’ Rule** $P[A_i|B] = \frac{P[B|A_i]P[A_i]}{P[B]}$
- **Expectation** If $X$ is a random variable, $E[X] = \sum x \cdot P[X = x]$.
- **Probabilistic Method** Compute the expected value of a random variable and from this deduce that there must be an outcome in the sample space where the random variable takes on a certain value (usually because the random variable takes on integer values, and the expected value computed is a non-integer, so the random variable must take on either a higher or lower value).
So I heard 3-COLOR is NP-complete

Suppose that \( n \geq 4 \) and let \( F \) be an arbitrary collection of \( n \)-subsets of \( E \). Suppose that \( |F| < \frac{4^{n-1}}{3^n} \). Show that there is a 4-coloring of \( E \) such that in every \( E_i \in F \), all 4 colors are represented.

Sketch: Randomly color the set \( E \) and define random variables for each set in \( F \) such that they are zero if the corresponding set contains all four colors and one otherwise. Prove that \( \mathbb{E}[X] < 1 \) and use the probabilistic method.

Not Enough Gold

The 251 TAs received 2 mysterious boxes with gold and silver coins. Box \( i \) has \( g_i \) gold coins and \( s_i \) silver coins, with \( g_i + s_i > 0 \) and \( i = 1, 2 \). Tim randomly picks a coin from box 1 and put it into box 2. Dmitriy then shuffles the boxes randomly. Mark then chooses one of the boxes and randomly picks a coin out from the box. What is the probability that he picks a gold coin?

Sketch: Use Law of Total Probability

Counting 1 2 3:

- Number of ways of arranging \( r \) out of \( n \) objects = \( \binom{n}{r} \) \!.
- Number of ways of choosing \( r \) out of \( n \) objects without repeats = \( \binom{n}{r} \).
- Number of ways to put \( m \) distinct balls into \( n \) distinct boxes = \( n^m \).
- Number of ways to distribute \( n \) indistinguishable gold among \( k \) distinct pirates = \( \binom{n+k-1}{k-1} \).
- Pigeonhole Principle Given \( n \) pigeons and \( k \) holes, then there exists a hole containing \( \geq \lceil \frac{n}{m} \rceil \) pigeons.

- Counting in Two Ways Make up stories to relate the LHS to the RHS.
- Principle of Inclusion-Exclusion

\[
\bigg| \bigcup_{i=1}^{n} A_i \bigg| = \sum_{i=1}^{n} |A_i| - \sum_{i,j:1 \leq i \leq j \leq n} |A_i \cap A_j| + \sum_{i,j,k:1 \leq i \leq j \leq k \leq n} |A_i \cap A_j \cap A_k| - \ldots + (-1)^{n-1} |A_1 \cap \ldots \cap A_n| 
\]

And You Thought You Hated Ducks

The ducks are back! Help Ambassador Amitriy to count the number of ways of placing \( m \) distinguishable ducks into \( n \) distinguishable ponds so that no pond contains more than \( B \) ducks. You are allowed to use three (!) summations. (Hint: You might want to use the summation \( \sum_{a_1 + \ldots + a_k = i} \) to denote summing over all possible combinations of \( a_j, 1 \leq j \leq k \) that sum to \( i \))

Sketch: The solution is

\[
\sum_{k=1}^{m} \left( -1 \right)^k \binom{n}{k} \sum_{i=0}^{m-kB'} \sum_{a_1 + \ldots + a_k = i} \binom{m}{kB' + i} \left( B' + a_1, \ldots, B' + a_k \right) (n-k)^{m-kB'-i} 
\]

Figure out the counting argument.
Practice Makes Perfect

Let a sequence of $n$ integers be given. Show that some consecutive subsequence has the property that the sum of the members of the subsequence is a multiple of $n$.

Sketch: Same idea as Target Practice. Define prefixes $a_1, \ldots, a_{n-1}$ and do pigeonhole.

So I Got Bored...

Ankur adores aadvarks, antelopes and alligators. (Okay, I give up the alliteration). Consider now the sequences of length $n$ consisting of \{aadvarks, antelopes, alligators\}. We say that sequences $a = a_1, a_2, \ldots, a_n$ and $b = b_1, b_2, \ldots, b_n$ collide if $a_i = b_i$ for some $i$. We say that a set $S$ is a colliding set if every pair $a, b \in S$ collide. What is the largest possible size of $S$? (Hint: bracketing might help)

Sketch: The answer is $3^{n-1}$. Partition the set into sets of size threes to get the upper bound. Just fix one element and vary the rest to find the lower bound.