

CSE 150 Fall 2009: Homework #3

Chao Xu

Problem 1

Set 1	Set 2	Any function	Injection	Surjection	Bijection
$f(f^{-1}(Y))$	Y	\subseteq	\subseteq	$=$	$=$
$f^{-1}(f(X))$	X	\supseteq	$=$	\supseteq	$=$
$f(f^{-1}(f(X)))$	$f(X)$	$=$	$=$	$=$	$=$
$f^{-1}(f(f^{-1}(Y)))$	$f^{-1}(Y)$	$=$	$=$	$=$	$=$
$f(X_1 \cup X_2)$	$f(X_1) \cup f(X_2)$	$=$	$=$	$=$	$=$
$f(X_1 \cap X_2)$	$f(X_1) \cap f(X_2)$	\subseteq	$=$	\subseteq	$=$
$f^{-1}(Y_1 \cup Y_2)$	$f^{-1}(Y_1) \cup f^{-1}(Y_2)$	$=$	$=$	$=$	$=$
$f^{-1}(Y_1 \cap Y_2)$	$f^{-1}(Y_1) \cap f^{-1}(Y_2)$	$=$	$=$	$=$	$=$

Problem 2

A vertex with even degrees is called a even-vertex, else it's a odd-vertex. Let's call a connected graph G with n edges and at most 2 odd-vertices n -euler, with all even-vertices n -even.

Lemma 1. *There is only even number of odd-verticies in every graph.*

Proof. Trivial from

$$\sum_{v \in V} \deg v = 2|E|$$

.

□

Lemma 2. *If there exist a Eulerian path for a n -euler graph's with odd-vertices, the path must start from an odd vertex.*

Proof. Two adjacent edges in a path shares a vertex. Thus if a vertex(not on the start or end position) appears k times in the path, then there are $2k$ edges containing that vertex. The only vertex that is not shared by two adjacent edges has to be in the start or end position, which has to be odd-vertices. □

Lemma 3. *If there exist a Eulerian path for a n -euler graphs with odd-vertices, the path must start and end in different vertices.*

Proof. From the proof of the above lemma, we can replace the first odd vertex with another. Then it proves both odd vertex need to be either in the start or the end vertex. Thus if one is the start vertex, the other must be the end vertex. Which are different vertices. □

Proof that if and only if G is a n -euler graph, then it have a Eulerian path

Let $P(n)$ be the predicate

All n -euler graphs have Eulerian paths AND all non n -euler graphs with n edges does not.

Base case: $P(1)$ is true.

Inductive step: Suppose $P(n)$ is true, then $P(n + 1)$ is true:

For the connected graph G with $n + 1$ edge to have a Eulerian path, one can pick a initial vertex and go though a edge and remove it. That step reduce the graph to G' with n edges. G' must be n -euler for G to contain Eulerian path. If not, there is no way to go though all the edges of G' exactly once; therefore there is no path going though all the edge of G .

Removal of a edge decrease two verticies's degree by 1.

If G is not a $n + 1$ -euler graph, then G can't have more than 4 odd-vertices, because G' is a n -euler graph. Thus G can only have 4 odd-vertices. There exist at least 1 pair of odd-vertices that have common edges. Removing the edge between those two vertices, we have a n -euler graph with odd-vertices. However, the starting vertex is on the edge we just removed, which is now a even-vertices. From the lemma, n -euler graph with odd-vertices can't have Eulerian paths starting from a even edge.

Thus *if G is not a $n + 1$ -euler graph, it doesn't have Eulerian paths.*

If G is a $n + 1$ -euler graph, there are 2 cases.

Case 1: G is a $n + 1$ -even graph.

Removal of any edge result a G' that is n -euler graph, thus having a Eulerian path. If the edge removed is $\{u, v\}$, the Eulerian path for G' contains v start vertex. Thus the edge removed can be appended in the end of the path and result a Eulerian path for G .

Case 2: G is not a $n + 1$ -even graph.

Remove any edge that contains at least 1 odd vertex, the result G' is a n -euler graph. The rest of the argument is similar to the case above

Thus *if G is a $n + 1$ -euler graph, then G have a Eulerian path.*

Hence $P(n)$ is true for all $n \geq 1$

Proof that if and only if G is a n -even graph, then G have a Eulerian cycle

$P(n)$ is true, and n -euler graph with 2 odd-vertices have different start and end vertex. Then n -euler graph with odd-vertices doesn't have Eulerian cycle.

Hence only n -even graphs can have Eulerian cycles.

Let G be a $n + 1$ -even graph. Select any vertex v , and let it be the starting vertex. Remove a edge that contains $\{v, u\}$ result a G' that is a n -euler graph and start at odd vertex u . It have a Eulerian path that end in v , since it's now a odd-vertex. Put back the edge $\{v, u\}$, we have the Eulerian cycle that begin and ends with v . This doesn't work where there are only 2 vertices, thus doesn't work for less than 3 edges. Hence there exist a Eulerian cycle for all n -even graphs($n \geq 3$).

Problem 3

Other than $f_8(n)$, other ones are trivial and can be evaluated either by taking the leading coefficient of the polynomial or the function itself with constant removed. $f_7(n)$ can be approximated by a integral.

$f_8(n)$ and $f_3(n)$ are asymptotically equivalent.

$f_4(n)$ and $f_7(n)$ are asymptotically equivalent.

$$f_8(n) = \binom{n}{k} = \frac{n!}{(n-k)!k!} = \frac{\prod_{i=0}^{k-1} (n-i)}{k!} = \Theta(n^k)$$

$$\begin{aligned} f_1(n) &= \Theta((\log n)^{\log n}) \\ f_2(n) &= \Theta(2^{\sqrt{n}}) \\ f_3(n) &= \Theta(n^k) \\ f_4(n) &= \Theta(n^{k+1}) \\ f_5(n) &= \Theta((\log n)^{\log \log n}) \\ f_6(n) &= \Theta(2^n) \\ f_7(n) &= \Theta(n^{k+1}) \\ f_8(n) &= \Theta(n^k) \end{aligned}$$

Only $\Theta((\log n)^{\log n})$ and $\Theta((\log n)^{\log \log n})$ are not trivial in comparison.

$\sqrt{n} > \log n \log \log n > \log n$, for sufficiently large $n \implies \Theta(2^{\sqrt{n}}) > \Theta(\log n^{\log n}) > \Theta(n^k)$ for any k .

$\log n > (\log \log n)^2$ for sufficiently large $n \implies \Theta(n^k) > \Theta((\log n)^{\log \log n})$.
Then.

$$f_5(n), f_3(n), f_8(n), f_4(n), f_7(n), f_1(n), f_2(n), f_6(n)$$

Problem 4

T is a set of all theorems that has a set of proof. $P(t)$ is the set of proof for theorem $t \in T$. let $O = \bigcup_{n=1}^{\infty} S^n$ then $P(t) \subseteq O$ O is a countable set, it is possible to create a bijection $f : \mathbb{N} \rightarrow O$. (For example, take any string made up of S be a number in base- $|S|$ representation) Let there be a function $C(n, t) : (\mathbb{N}, T) \rightarrow \{0, 1\}$, such that

$$C(n, t) = 1 \Leftrightarrow f(n) \in P(t)$$

$$C(n, t) = 0 \Leftrightarrow f(n) \notin P(t)$$

We assume $C(n, t)$ and $f(n)$ is computable. The following algorithm will find such proofs, and it will halt for any $t \in T$ because there exist a n such that $f(n) \in P(t)$

```

procedure return-proof( $t$ )
   $n \leftarrow 0$ 
  while ( $C(n, t) \neq 1$ )
     $n \leftarrow n + 1$ 
  return  $f(n)$ 

```

Thus the proof can be found eventually.

Problem 5

We need a function that that $A[B[i]] = i$ for all $0 \leq i < n$

```

procedure inv( $A[0, \dots, n - 1]$ )
   $i \leftarrow 0$ 
  while  $i < n$ 
     $R[A[i]] \leftarrow i$ 
     $i \leftarrow i + 1$ 
  return  $R$ 

```

```

   $B = \text{inv}(A)$ 
   $B[A[i]] = i$ 
   $A[B[A[i]]] = A[i]$ 
  let  $k = A[i]$ 
   $A[B[k]] = k$ 

```

Thus prove the function works.

Problem 6

```

procedure find-kth-smallest( $A, n, k$ )
   $p \leftarrow \text{partition}(A[0, \dots, n - 1], A[0])$ 
  if  $p = k$ 
    return  $A[0]$ 
  swap( $A[0], A[p - 1]$ )
  if  $p > k$ 
    return find-kth-smallest( $A[0, \dots, p - 1], p, k$ )
  if  $p < k$ 
    return find-kth-smallest( $A[p + 1, \dots, n - 1], n - p - 1, k - p - 1$ )

```

During the analysis of quicksort, we have the best case of quicksort follows this relation

$$T(n) = \Theta(n) + 2T\left(\frac{n}{2}\right)$$

The above algorithm matches quicksort, but with only 1 recursive call. Thus we have

$$T(n) = \Theta(n) + T\left(\frac{n}{2}\right)$$

Then $T(n)$ is the average time complexity for $\text{find-kth-smallest}(A, n, k)$. Let $g(n) = \Theta(n)$.

There exist a $\epsilon = 1$, such that $g(n) = \Omega(n^{\log_2(1)+\epsilon})$

There exist $c < 1$, such that $g\left(\frac{n}{2}\right) \leq cg(n)$.

From the master's theorem, $T(n) = \Theta(\Theta(n)) = \Theta(n)$.