Design and Analysis of Algorithms
End-Semester Examination: Marking scheme

November 28, 2012

1. Failure probability of a path $P = 1 - \prod_{e \in P}(1 - p_e)$. Minimizing this is equivalent to maximizing $\prod_{e \in P}(1 - p_e)$, hence maximize $\sum_{e \in P} \log(1 - p_e)$, hence minimize $\sum_{e \in P} \log \frac{1}{1-p_e}$. With these edge weights, execute Dijkstra's algorithm. As $0 \leq p_e < 1$, $0 < 1 - p_e \leq 1$.

   Correct algo: 4 marks, complexity: 1 mark

2. Maintain pointers to first occurrences of a ball of each color. Hence you need 3 pointers. Swap the balls pointed to by the pointers, if necessary and modify the poiners to again point to the correct position. Only red pointer needs to point to the first *misplaced* red ball.

   You need to prove that for each pointer, the array is scanned only once throughout the algorithm. Hence $O(n)$ time.

   Correct algo: 4 marks, complexity+correctness: 3 marks

3. Maintain a sorted array B of distinct elements. Scan the original array A, each time insert an element in B at correct position if it's not already present.

   Checking an element's presence in B takes $O(\log \log n)$, insertion takes $O(\log n)$. Total time for search=$O(n \log \log n)$, total time for insertion= $\log^2 n$.

   Correct algo: 5 marks, analysis: 3 marks

4. An Euler traversal gives the required list. Each edge is traversed twice, so $O(n)$ time.

5. (a) Case 1: $A = \{s\}$: Trivially no edges between A and $V \setminus A$.

   Case 2: $B = \{t\}$: Again trivial

   Case 3: $V \cap A = \emptyset$: This is a cut in which $A \setminus \{s\} \subseteq U$. Thus the edges cut are from s to $U \setminus A$ and from A to V. You can argue that there is another cut of the same cardinality or less if you A so that it becomes $N(A) \cup A$. Of course $N(A) \subseteq V$ and you reach one of the other cases now.

   Case 4: $V \cap A \neq \emptyset$. Let $X = U \cap A$ and $Y = V \cap A$. Also, let $X' = U \setminus X, Y' = V \setminus Y$. Thus $X', Y' \subset B$. $|Cut| = |X'| + |Y| + |E(X, N(X))|$.

   As earlier, if $N(A) \cap X' \neq \emptyset$, modify the partitions by putting $N(X)$ into A. In terms of the sets described above, the new cut size is $|X'| + |Y| + |N(X)|$. As $|E(X, N(X))| \geq |N(X)|$, this cut is at most as large as the previous one and has no edges from A to $V \setminus A$.

   (Ideally, I should have drawn figures here but I am not very good at it.)

   Cases 1,2: 1 mark each, Case 3: 2 marks, Case 4: 3 marks

   (b) Required vertex cover is $X' \cup N(X)$. (Note that vertex cover is required for original graph, not for the flow graph. Hence you need not cover s to A edges.

   (c) $A = \{v_1, v_2, v_3, u_1, u_2, s\}$, B=rest.

(d) $\exists$ perfect matching $\Leftrightarrow$ maxflow$=n$. If there is a subset X such that $|X| > |N(X)|$, form a cut as $A = \{s\} \cup X \cup N(X)$ and B the rest. This cut has size $(n - |X|) + |N(X)| < n$. Converse can be proved similarly.

First implication: 1 mark, above argument including converse: 3 marks

6. (a) Decision version, containment in NP : 1 mark

Reduction: Given an instance of set cover problem, sets form X partition in the bipartite dominating set instance, elements in the universe form Y partition. Correctness is trivial. 3 marks.

(b) Form clique on X to get an instance G' of dominating set problem. For each dominating set that includes vertices in Y, remove those vertices and put their neighbours from X into the dominating set. Argue that this does not increase the size. Thus G' has a dominating set of size k iff G does.

(c)

$$
\begin{aligned}
&Minimize \sum_{v \in V} x_v \quad subjectto \\
x_v + \sum_{u:(u,v)\in E} x_u \quad &\geq \quad 1 \forall v \in V \\
x_v \quad &\in \quad \{0,1\} \forall v \in V
\end{aligned}
$$

(d) (i) In any feasible solution of the relaxed LP, at least one of the $d+1$ variables in each constraint has value $\geq \frac{1}{d+1}$.

(ii) Opt of LP $\leq$ Opt of ILP: 2 marks The above modification scales a variable by at most a factor of $d+1$. Hence the algorithmic solution $\leq (d+1)$(LP opt)$\leq (d+1)$ILP opt : 2 marks

7. (a) Any $k+1$ clique can not be colored in k colors.

(b) A random coloring leaves an edge unsatisfied with probability $\frac{1}{k}$: 2 marks

Define a random variable X=number of satisfied edges in a random coloring. Define random variables X_e such that $X_e = 1$ if the coloring satisfies e and $X_e = 0$ otherwise. $E[X] = \sum_{e \in E} E[X_e] = \frac{k-1}{k}|E|$: 2 marks

Hence there exists required coloring: 2 marks

(c) Use method of conditional expectation as we used in class for 3-SAT. Pick a vertex v_1 and split the expectation into k sums as

$$E[X] = \sum_{i=1}^{k} \frac{1}{k} E[X|color(v_1) = i]$$

. By averaging argument, there exists one value of i where the conditional expectation is at least $\frac{k-1}{k}|E|$.