University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
P. Bamberger

Algorithm Theory
Exercise Sheet 8

Exercise 1: Online Vertex Cover (*Points*)

Let $G = (V, E)$ be a graph. A set $S \subseteq V$ is called a *vertex cover* if and only if for every edge $\{u, v\} \in E$ at least one of its endpoints is in S. The minimum vertex cover problem is to find such a set S of minimum size.
We are considering the following online version of the minimum vertex cover problem. Initially, we are given the set of nodes V and an empty vertex cover $S = \emptyset$. Then, the edges appear one-by-one in an online fashion. When a new edge $\{u, v\}$ appears, the algorithm needs to guarantee that the edge is covered (i.e., if this is not already the case, at least one of the two nodes u and v needs to be added to S). Once a node is in S it cannot be removed from S.

(a) Provide a deterministic online algorithm with competitive ratio at most 2. That is, your online algorithm needs to guarantee at all times that the vertex cover S is at most by a factor 2 larger than a current optimal vertex cover. Prove the correctness of your algorithm.

(b) Show that any deterministic online algorithm for the online vertex cover problem has competitive ratio at least 2.

(c) Use Yao's principle to show that any randomized online algorithm for the online vertex cover problem has competitive ratio at least 3/2.

Exercise 2: Ski Rental Problem (*Points*)

Assume you go skiing and have to decide whether to buy or rent skis. Assume that cost is your only concern and that buying ski is k times more expensive than renting it for a day.
After buying your own ski you can use them forever without additional cost. Finally assume that you have no knowledge how many days n you will ever go skiing in total.
This means that as long as you go skiing you have to decide on every new skiing-day in an online fashion whether to buy or rent, until you finally decide to buy a pair of ski.

(a) Describe the best offline strategy OPT (n is known beforehand) and give the cost as a function depending on n.

(b) Assume your online strategy ALG (where you have no knowledge of n) is to buy a pair of ski before your first skiing day. Give a lower bound for the strict competitive ratio of ALG and explain it briefly.

(c) Give a strategy ALG' that is strictly 2-competitive and prove it.

Exercise 3: Maximum Cut *(Points)*

Let $G = (V, E)$ be an arbitrary unweighted undirected graph. A maximum cut of G is a cut whose size is at least the size of any other cut in G.

(a) ***(4 Points)*** Give a simple randomized algorithm that returns a cut of size at least 1/2 times the size of a maximum cut *in expectation* and prove this property.

(b) ***(10 Points)*** Prove that the following deterministic algorithm (Algorithm 1) returns a cut of size at least 1/2 times the size of a maximum cut.

Algorithm 1 Deterministic Approximate Maximum Cut

```
Pick arbitrary nodes v_1, v_2 ∈ V
A ← {v_1}
B ← {v_2}
for v ∈ V \ {v_1, v_2} do
    if deg_A(v) > deg_B(v) then          ▷ deg_X(v) is the number of v's neighbors in X ⊆ V.
        B ← B ∪ {v}
    else
        A ← A ∪ {v}
Output A and B
```

(c) ***(5 Points)*** Let us now consider an online version of the maximum cut problem, where the nodes V of a graph $G = (V, E)$ arrive in an online fashion. The algorithm should partition the nodes V into two sets A and B such that the cut induced by this partition is as large as possible. Whenever a new node $v \in V$ arrives together with the edges to the already present nodes, an online algorithm has to assign v to either A or B. Based on the above deterministic algorithm (Alg. 1), describe a deterministic online maximum cut algorithm with *strict competitive ratio* at least 1/2. You can use that fact that Algorithm 1 computes a cut of size at least half the size of a maximum cut.

Hint: An online algorithm for a maximization problem is said to have strict competitive ratio α if it guarantees that $\text{ALG} \geq \alpha \cdot \text{OPT}$, *where* ALG *and* OPT *are the solutions of the online algorithm and of an optimal offline algorithm, respectively.*

(d) ***(7 Points)*** Show that no deterministic online algorithm for the online maximum cut problem can have a strict competitive ratio that is better than 1/2.