Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

HS 2021

Prof. R. Wattenhofer
Peter Belcak

Computational Thinking
Exercise 4

1 Bicolored Edges

Let us consider the Bicolored Edges problem: given an input graph $G = (V, E)$, our job is to color the nodes of G with two colors such that the number of edges with different-colored endpoints is as large as possible.

Given a current coloring, let us call a node v with current color c_v a *wasteful* node if it has more neighbors of color c_v then of the opposite color. In this case, changing the color of v to the opposite color would improve our current solution. This suggests the following greedy algorithm:

```
def Bicolored_Greedy(G):
    begin with an arbitrary coloring of V
    while there is a wasteful node v:
        change the color of v to the opposite color
    return the current coloring
```

a) Find an example graph where this algorithm might return a coloring that is worse than the optimum!

b) Prove that the main loop of the algorithm is repeated at most $O(n^2)$ times before the algorithm terminates, where $n = |V|$.

c) Show that this greedy algorithm is a 2-approximation for Bicolored Edges.

2 Finding 4-segments

Given a graph $G = (V, E)$, a 4-*segment* is a path consisting of 4 edges, i.e. distinct nodes v_1, v_2, v_3, v_4, v_5 such that $(v_1, v_2), (v_2, v_3), (v_3, v_4), (v_4, v_5) \in E$. We say that two 4-segments are *disjoint* if they do not share a common edge. Note that disjoint 4-segments can still share a common node. Our goal is to find the highest number of 4-segments in G that all are pairwise disjoint.

Assuming we already have a set S of selected 4-segments, we say that a 4-segment s in G is *free* if s is disjoint from every 4-segment in S. Now consider the following greedy algorithm:

```
def Find4segments_Greedy(G):
    S = ∅
    while there exists a free 4-segment s:
        S = S ∪ {s}
    return S
```

Prove that this algorithm is a 4-approximation for the problem.