HKOI Final 2014 Solution
Senior Q1: Dividing the Cities

January 11, 2014

Problem statement

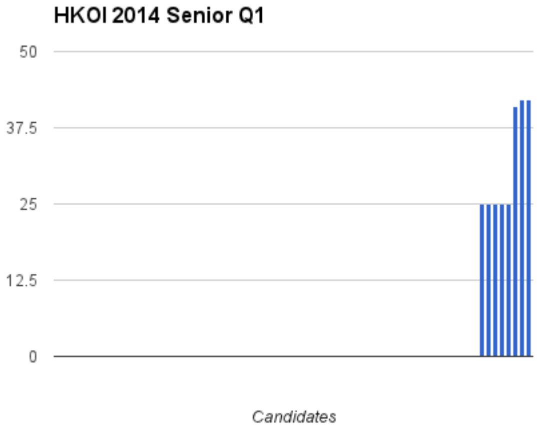
- Alex gets graph G and a valid 10-coloring of G as input. He processes the input and output a 01-string s.
- Bob gets G and s as input. He has to output a valid 10-coloring for G. Notice that it is **not** required to be same as Alex's one.
- Your task is to design a "protocol" for them, and write two programs to simulate their behaviour.

Problem statement

- Alex gets graph G and a valid 10-coloring of G as input. He processes the input and output a 01-string s.
- Bob gets G and s as input. He has to output a valid 10-coloring for G. Notice that it is **not** required to be same as Alex's one.
- Your task is to design a "protocol" for them, and write two programs to simulate their behaviour.
- The shorter s, the higher score.
- $|V| \leq 3000, |E| \leq 10000$
- $|s| \leq 4800$ for full mark.

Statistics

- Maximum $= 41.83$



Task analysis

- 10-coloring is a hard problem, no polynomial time algorithm has been discovered yet.
- Hence Bob must receive some help from Alex to solve the problem.

Basic solution

- Alex simply encodes the 10-coloring as s and send it to Bob
- Use 4 bits to encode the color of each vertex ($2^4 = 16 \geq 10$)
- Total bits used $= 3000 \times 4 = 12000$
- Expected score: 25

Optimization 1: Improved encoding

- Using 4 bits to encode 10 colors is inefficient
- In theory, $\log_2 10 \approx 3.32$ bits is sufficient
- Use 10 bits to encode 3 vertices ($2^{10} = 1024 \geq 10^3 = 1000$)
- Total bits used $= 3000/3 \times 10 = 10000$
- Expected score: 41.67

Optimization 2: Large degree vertex

Although 10-coloring is hard, do not forget that Bob still have some abilities for some calculation.

Optimization 2: Large degree vertex

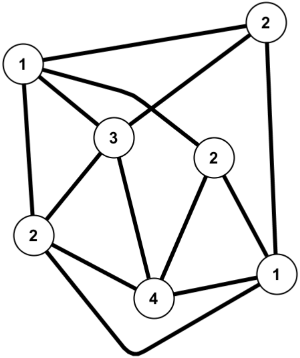
Although 10-coloring is hard, do not forget that Bob still have some abilities for some calculation.

- Let the *degree* of a vertex to be the number of edges connected to it
- If the degree of vertex is small, then it should be easier to color it
- For vertices with degree < 10, Bob can always find a proper color for it regardless on how other vertices is colored
- Alex send the color of vertices with degree ≥ 10 only

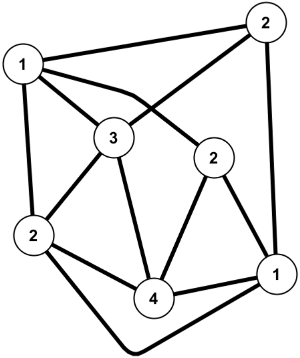
Optimization 2: Large degree vertex

- Each edge contributes 2 to the sum of degree of the whole graph, which equals $2|E| \leq 20000$
- Number of vertices with degree at least 10 $\leq 20000/10 = 2000$
- Total bits used $= \lceil 2000/3 \rceil \times 10 = 6670$
- Expected score: 69.41

Optimization 3: Bipartiteness

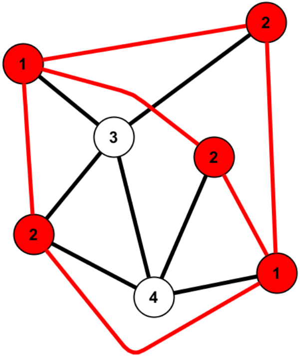


Optimization 3: Bipartiteness

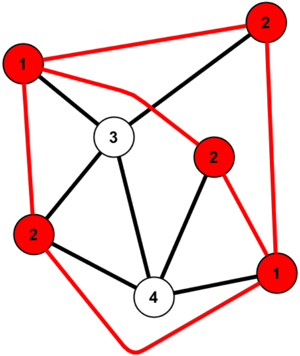


Now look closer to the vertices which is colored 1 or 2

Optimization 3: Bipartiteness



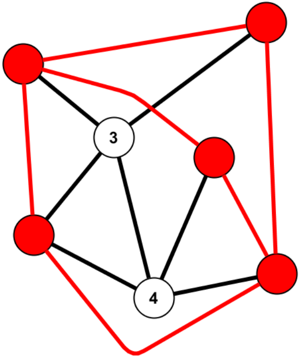
Optimization 3: Bipartiteness



The induced subgraph formed by it is **bipartite**!

Optimization 3: Bipartiteness

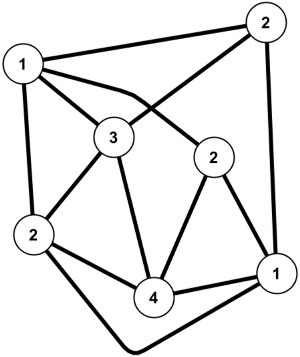
Color the vertex marked red using two colors $\{1, 2\}$, known as *bicoloring*, is easy.
It can be done in linear time by running a DFS/BFS.



Same for $\{3, 4\}, \{5, 6\}, \{7, 8\}, \{9, 10\}$

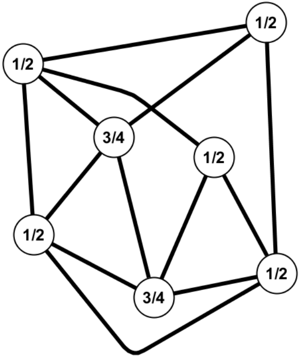
Optimization 3: Bipartiteness

Therefore, Alex instead of sending the explicit color (10 choices), he only have to tell Bob whether it is $\{1, 2\}$ or $\{3, 4\}$ or $\{5, 6\}$ etc. Bob can bicolor each set by himself.



Optimization 3: Bipartiteness

Therefore, Alex instead of sending the explicit color (10 choices), he only have to tell Bob whether it is $\{1,2\}$ or $\{3,4\}$ or $\{5,6\}$ etc. Bob can bicolor each set by himself.



Optimization 3: Bipartiteness

- Each vertex only get 5 choices
- Use 7 bits to encode 3 vertices ($2^7 = 128 \geq 5^3 = 125$)
- Total bits used $= \lceil 2000/3 \rceil \times 7 = 4669$
- Expected score: 100