HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

Algorithms and Data Structures

Laboratory work #7

Michael Kosyakov
Associate Professor

Denis Tarakanov
Assistant

Aglaya Iliina
Associate Professor

hduitmo.ads@yandex.ru

Classes plan

1. Previous homework problem #1650 “Billionaires”
2. Problem #1080 “Map Coloring”
3. Problem #1806 “Mobile Telegraphs”
4. Task for homework

Problem #1650 "Billionaires"

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Link to the problem's description https://acm.timus.ru/problem.aspx?space=1&num=1650&locale=en

- Your employer asks you to determine for each city the number of days during period of time on which this city exceeded all other cities in the total amount of money that billionaires staying in this city have.
- The N lines contain information about billionaires: names, cities where they were staying at the beginning of the period, and their fortunes. The K lines contain the list of travels: the number of days, name of the person, and city of destination.
- In each line of the output give name of a city and the number of days during which this city was the first with respect to the sum of fortunes of the billionaires staying there.

Problem #1650 “Billionaires”

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Implementation focused problem
- Restrictions allow to simulate the problem
- There is a lot of possible approaches

Problem #1650 “Billionaires”

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- What properties do different entities have?
- Billionaire
  - Name
  - Current city
  - Number of money
- City
  - Name
  - Current number of money in the city
  - Number of days when this city exceeded all other cities

Problem #1650 “Billionaires”

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Cities
  - map<name, city_properties>
  - Sorted by name, useful for output
- Billionaires
  - map<name, billionaire_properties>

- How to fast detect which city is leader?
  - set<pair<name, city_properties>, comparator>
  - Comparator should sort set by current amount of money in the city
  - Values inside set are constant!

Problem #1650 "Billionaires"

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- What should be done on each billionaire's travel?
- Step 1. Update city with maximum amount of money
  - How many days have passed since previous travel?
  - Current city with maximum amount of money was leader for this number of days
  - If there are 2 or more cities with same maximum amount of money – no leaders for this period!

City Name	Total money
City2	4000
City1	1000
City3	0

City Name	Total money
City2	4000
City1	4000
City3	0

Problem #1650 "Billionaires"

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Step 2. Update information
  - Find in which city billionaire was in previous period
  - Update information about billionaire with new destination

Input: 5 Billionaire2 City3

Billionaire	Cities	Money
Billionaire1	City1	1000
Billionaire2	City2	2000
Billionaire3	City2	2000

Billionaire	Cities	Money
Billionaire1	City1	1000
Billionaire2	City3	2000
Billionaire3	City2	2000

Problem #1650 "Billionaires"

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Step 2. Update information
  - Find this city in set, decrease its total money
  - Find destination city in set, increase its total money
  - After removing/inserting set should be resorted

Input: 5 Billionaire2 City3

City Name	Total money
City2	4000
City1	1000
City3	0

City Name	Total money
City1	1000

City2.totalMoney -= 2000
City3.totalMoney += 2000

Problem #1650 "Billionaires"

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Step 2. Update information
  - Find this city in set, decrease its total money
  - Find destination city in set, increase its total money
  - After removing/inserting set should be resorted

Input: 5 Billionaire2 City3

City Name	Total money
City1	1000
City2	2000
City3	2000

City Name	Total money
City2	2000
City3	2000
City1	1000

Problem #1650 “Billionaires”

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Step 3. Print result
  - Print all cities sorted by name
  - City should be a leader at least for one day to be printed
  - Don’t miss first period (from day 1 till first travel) and last period (from last travel till day M)

Problem #1080 “Map Coloring”

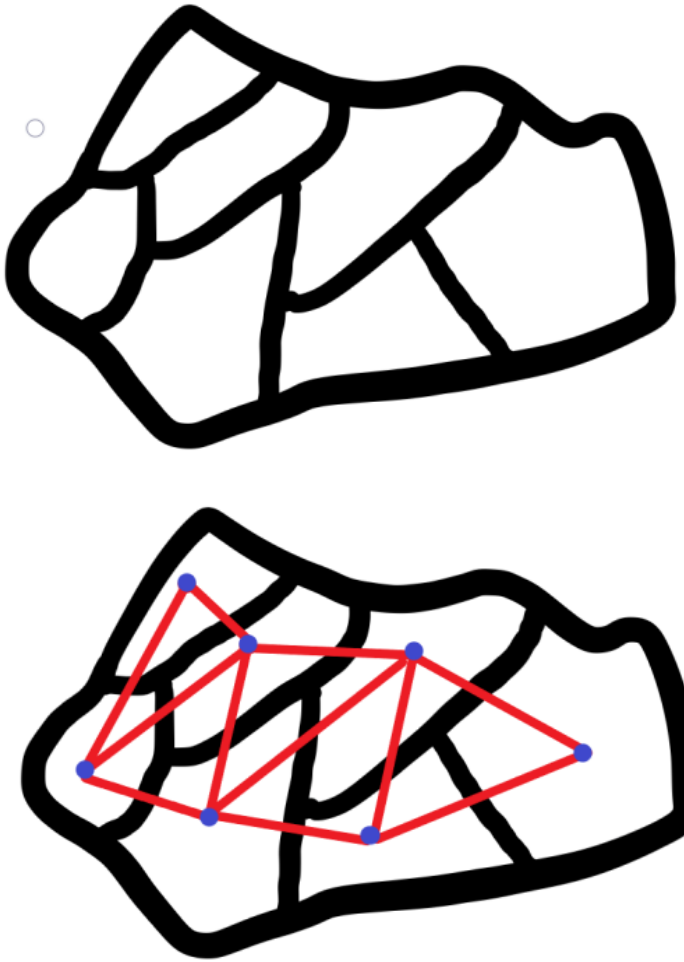
- Link to the problem's description https://acm.timus.ru/problem.aspx?space=1&num=1080&locale=en

- We consider a geographical map with N countries numbered from 1 to N. For every country we know the numbers of other countries which are connected with its border. From every country we can reach to any other one, eventually crossing some borders. Write a program which determines whether it is possible to color the map only in two colors — red and blue in such a way that if two countries are connected their colors are different. The color of the first country is red. Your program must output one possible coloring for the other countries, or show, that such coloring is impossible.

Problem #1080 “Map Coloring”

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

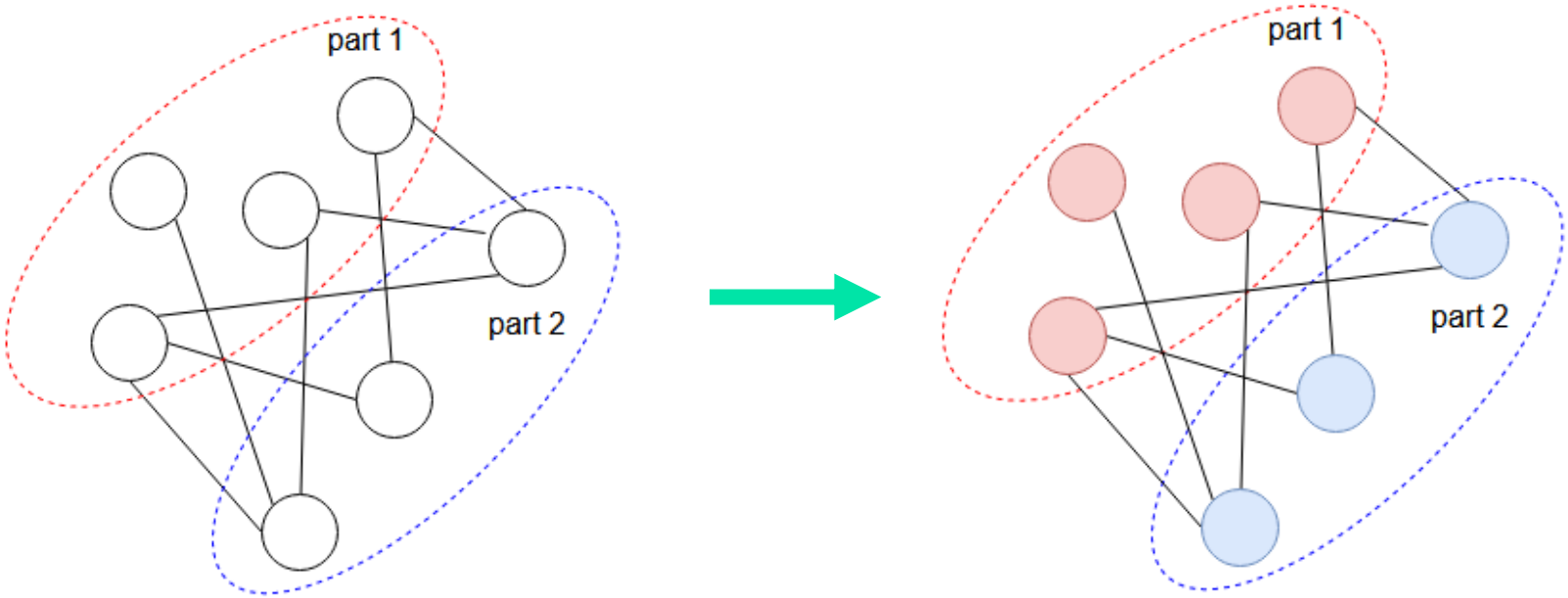
- Map of countries is a plane graph
- Let’s create a dual graph for this plane graph
- There is a path between each two countries – dual graph is connected



Problem #1080 “Map Coloring”

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

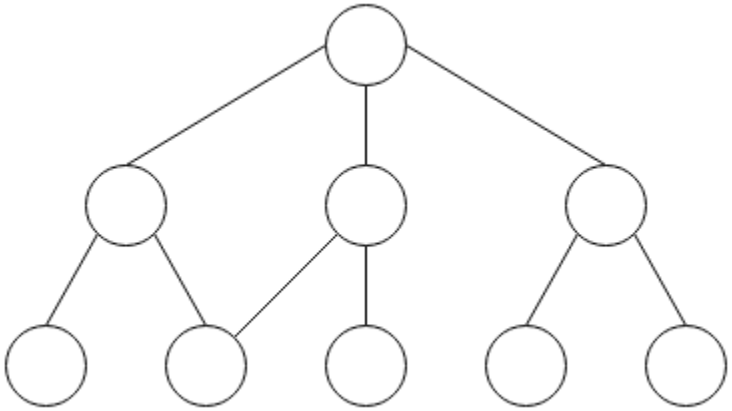
- Map coloring == proper vertex coloring of dual graph
- Bipartite graph
- Neighbor vertices have different color – each part has its own color

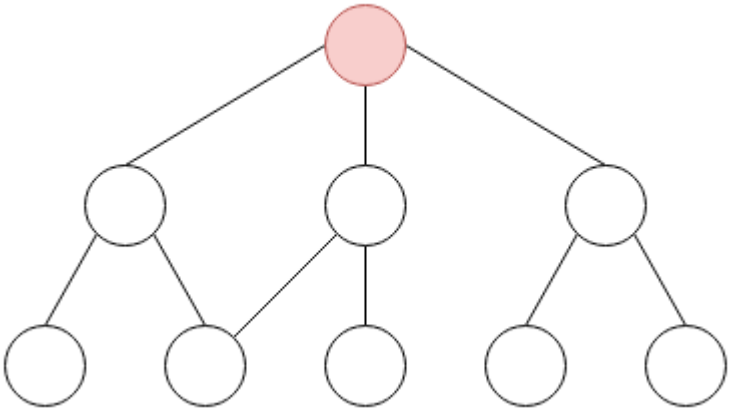


Problem #1080 “Map Coloring”

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

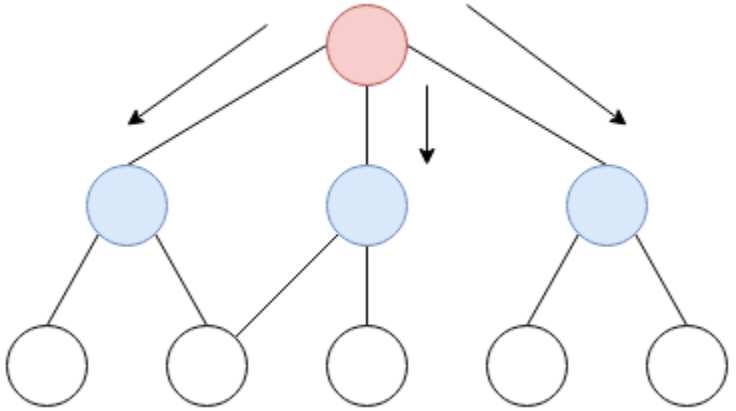
- DFS vs BFS
- BFS
- First country color is red

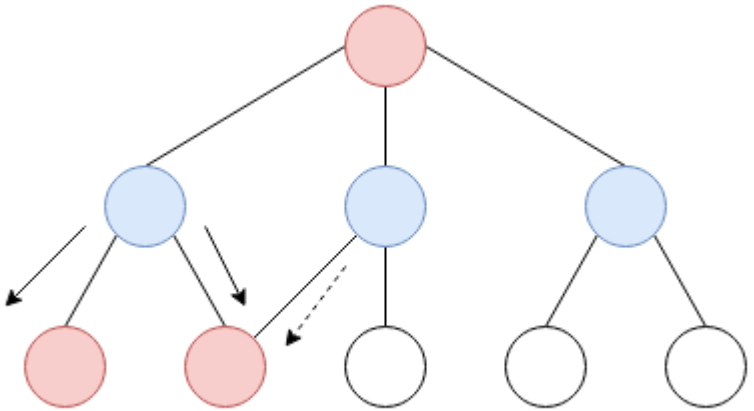




Problem #1080 “Map Coloring”

- Let’s color next level in blue
- All vertices have no colors
- Next level should be red

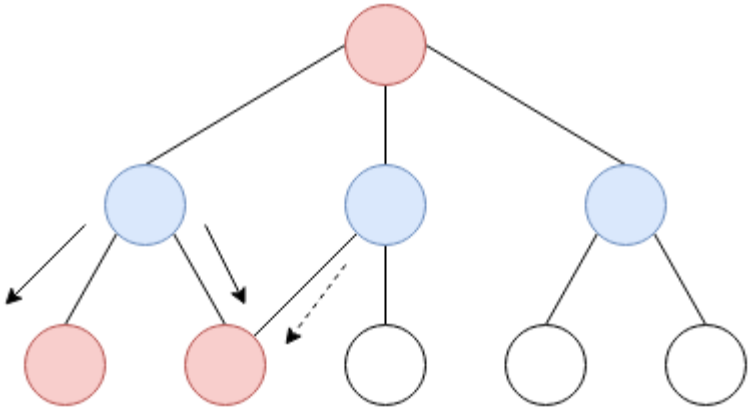


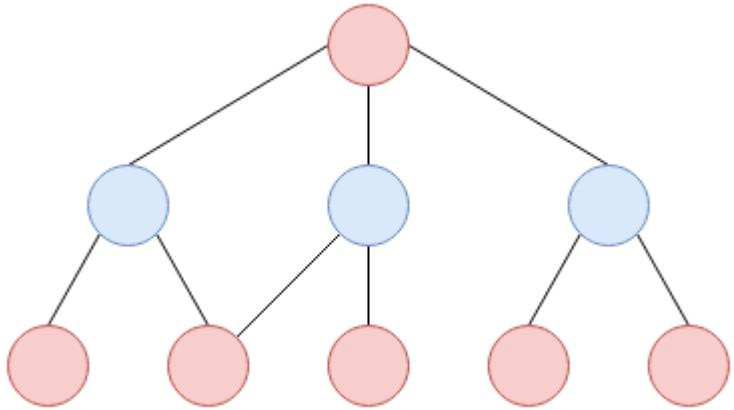


Problem #1080 “Map Coloring”

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

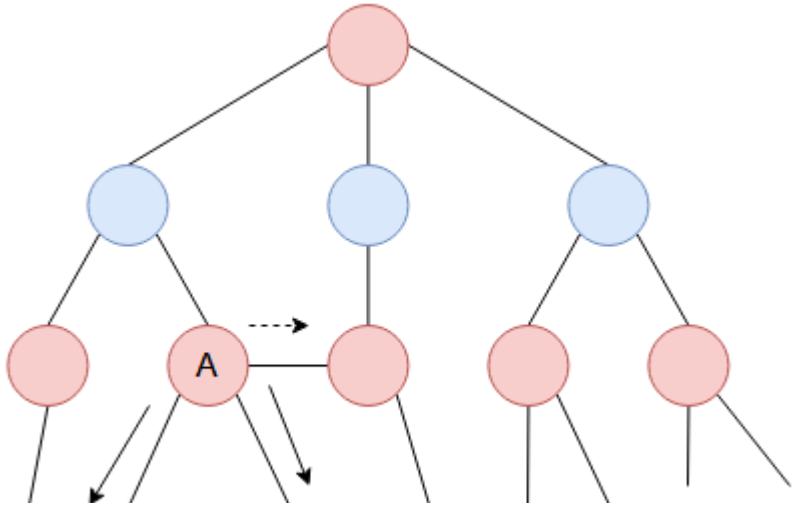
- Next level should be red
- Vertex already colored
- If it has “expected” color – we can continue, coloring still proper

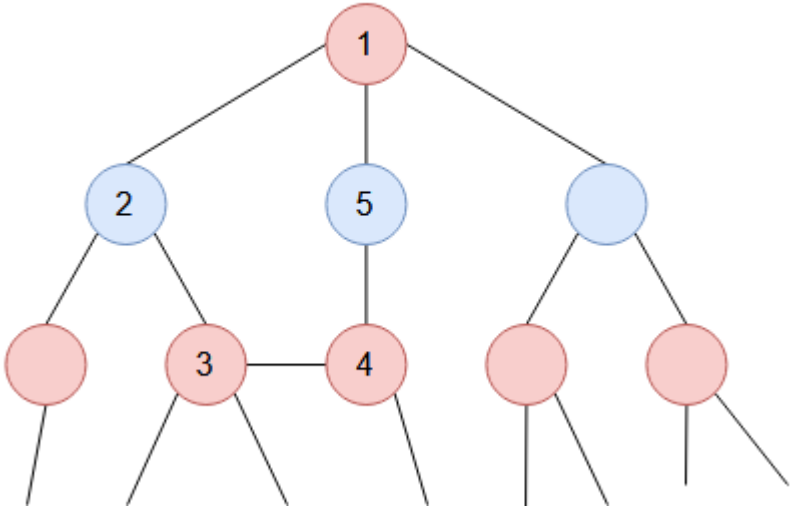




Problem #1080 "Map Coloring"

- Another example
- When we try to color neighbors of vertex A in blue – one of them already red
- Bipartite graph can't have odd cycles





Problem #1806 “Mobile Telegraphs”

- Link to the problem's description https://acm.timus.ru/problem.aspx?space=1&num=1806&locale=en

- Each device has a unique number, which is a string consisting of ten decimal digits. A message can be sent from a telegraph a to a telegraph b only if the number b can be obtained from the number a by changing exactly one digit or by swapping two digits, and the time of sending a message from the telegraph a to the telegraph b depends on the length of the longest common prefix of their numbers.
- Output “-1” if it is impossible to deliver the message. Otherwise, in the first line output the minimal time required to deliver the message. In the second line – number of fighters in the delivery path, and in the third line – their numbers

Problem #1806
“Mobile Telegraphs”

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Shortest path problem
- All weights are positive
- Fastest solution: Dijkstra’s algorithm with remembering shortest path
- How to build a graph?

Problem #1806 “Mobile Telegraphs”

- Memory limit
- Optimize graph presentation:

1. Total number of neighbor vertices is limited
2. Generate neighbor vertices on demand fast and in one place

Problem #1806 "Mobile Telegraphs"

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Total number of neighbor vertices is limited
  - Swapping of two digits

combination	9	1	2	3	4	9	3	3	4	2
1	1	9	2	3	4	9	3	3	4	2
2	2	1	9	3	4	9	3	3	4	2
3	3	1	2	9	4	9	3	3	4	2
4	4	1	2	3	9	9	3	3	4	2
					...					
33	9	1	2	3	3	9	3	4	4	2
					...					

  - $\frac{9*10}{2} = 45\ combinations$

Problem #1806 "Mobile Telegraphs"

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

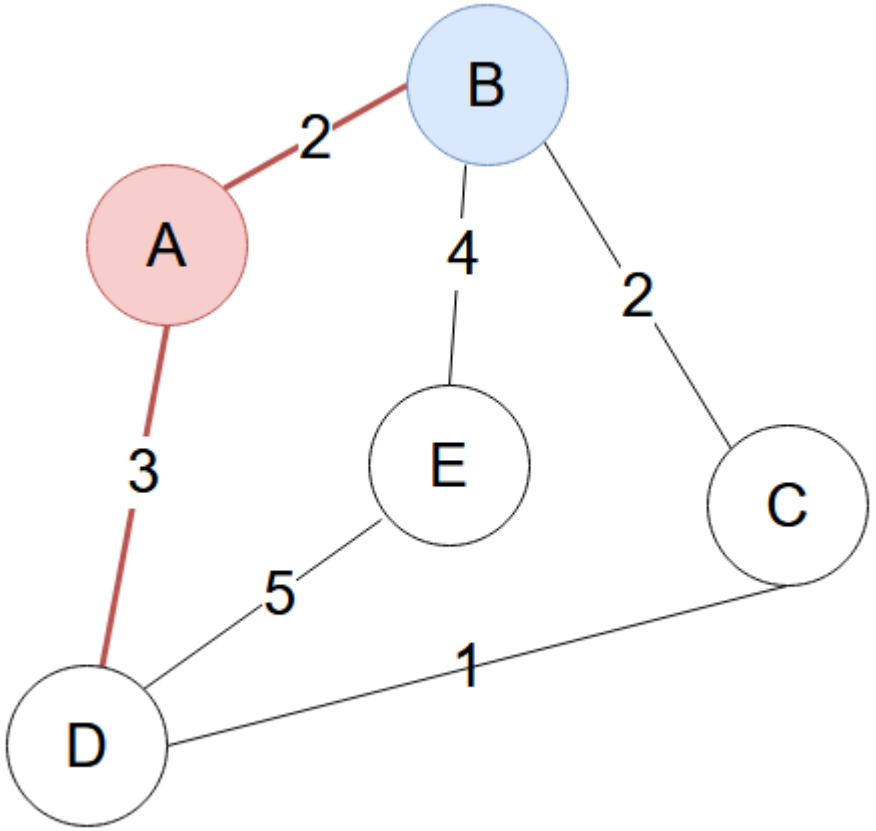
- Total number of neighbor vertices is limited
  - Changing one digit

combination	9	1	2	3	4	9	3	3	4	2
1	0	1	2	3	4	9	3	3	4	2
2	1	1	2	3	4	9	3	3	4	2
3	2	1	2	3	4	9	3	3	4	2
				...						
28	9	1	2	0	4	9	3	3	4	2
				...						

  - $9 * 10 = 90\ combinations$
  - Total: 90 + 45 = 135 possible neighbor vertices

Problem #1806 "Mobile Telegraphs"

- Generate neighbor vertices on demand fast and in one place
  - When should Dijkstra's algorithm know neighbor vertices for vertex B?
  - Relaxation of edges for this vertex



Problem #1806 “Mobile Telegraphs”

- Generate neighbor vertices on demand fast and in one place
  - Generate only all possible 135 neighbors
    - Change one digit for each position
    - Swap each pair of two digits
  - Check, if such sequence is presented as available telegraph number (hash map)
  - Calculate weight (length of prefix) for existed edges

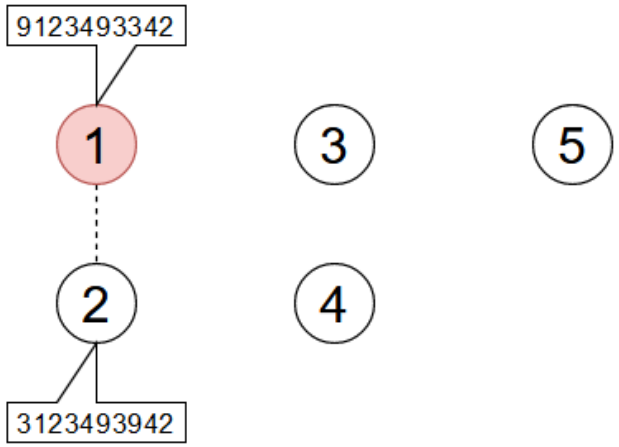
Problem #1806 "Mobile Telegraphs"

- Generate neighbor vertices on demand fast and in one place
  - No need to create 135 new strings each time
  - Use fix-sized array of strings, and update strings inside it
  - N.B. Language specific difference
    - For C++ std::string is mutable, strings can be changed with operator[]
    - For Java String is immutable, but can be replaced with char[135][10] array for all combinations, and build as string with constructor String(char[])

Problem #1806 “Mobile Telegraphs”

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

Current graph visualization



Hash map of all telegraphs

Telegraph (key)	Vertex (value)
9123493342	1
3123493942	2
9223433942	3
3223493942	4
9223433945	5

Array of possible neighbors of current vertex

Type	9	1	2	3	4	9	3	3	4	2	Found in map?
Swap combin.	1	9	2	3	4	9	3	3	4	2	Not found
	…										
	3	1	2	3	4	9	3	9	4	2	Found – vertex 2
	…										
Change combin.	0	1	2	3	4	9	3	3	4	2	Not found
	…										

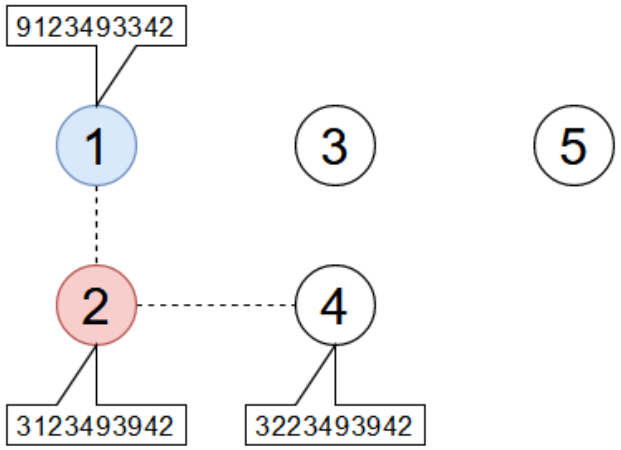
Problem #1806 "Mobile Telegraphs"

- What is weight of found edges?
  - Calculate length of common prefix
  - Get weight for this prefix length
- Usual Dijkstra's algorithm actions
  - Calculate DS to all found neighbors
  - For each neighbor check if new DS to vertex is less than known value
    - Add neighbors with their scores to priority queue or update score if vertex is already in a queue
    - Store new shortest paths to neighbors
  - Mark current vertex as vertex with already found shortest path and get from priority queue next vertex with minimal DS

Problem #1806 “Mobile Telegraphs”

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

Current graph visualization



Hash map of all telegraphs

Telegraph (key)	Vertex (value)
9123493342	1
3123493942	2
9223433942	3
3223493942	4
9223433945	5

Array of possible neighbors of current vertex

Type	3	1	2	3	4	9	3	9	4	2	Found in map?
Swap combin.	…										
	9	1	2	3	4	9	3	3	4	2	Found – vertex 1
	…										
Change combin.	…										
	3	2	2	3	4	9	3	9	4	2	Found – vertex 4
	…										

Mandatory task

1. Prepare source code to solve problem #1080 “Map Coloring” https://acm.timus.ru/problem.aspx?space=1&num=1080&locale=en
2. Pass tests on Timus system for this problem https://acm.timus.ru/submit.aspx?space=1&num=1080
3. Prepare a report with algorithm complexity and explanation
   Use template.docx to prepare report and send it to hduitmo.ads@yandex.ru with correct subject

Task for homework

You can solve following problems to get extra 2 points for each problem:

1. Problem #1806 "Mobile Telegraphs"
   https://acm.timus.ru/problem.aspx?space=1&num=1806&locale=en
   Solution of this problem was already explained
2. Problem #1450 "Russian Pipelines"
   https://acm.timus.ru/problem.aspx?space=1&num=1450&locale=en
   Report for this problem should contain analysis of graph type and comparison of possible solutions

Thank you!