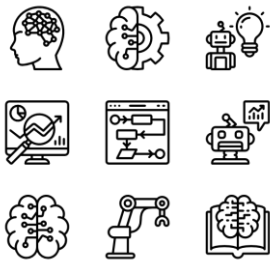


## Computer Science for Practicing Engineers

# Phương pháp tham lam



TS. Huỳnh Bá Diệu  
Email: dieuhb@gmail.com  
Phone: 0914146868

1

## Phương pháp tham lam (greedy)

---

### Nội dung:

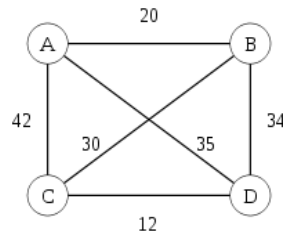
1. Tại sao phải dùng thuật toán tham lam?
2. Ưu điểm của thuật toán tham lam
3. Hạn chế
4. Một số bài toán sử dụng thuật toán tham lam

2

## Phương pháp tham lam: Bài toán người du lịch

Cho  $n$  thành phố, giữa các thành phố có các chuyến bay nối với nhau.

Một người du lịch xuất phát từ thành phố A, ông ta muốn qua tất cả các thành phố, mỗi thành phố chỉ qua 1 lần, rồi quay lại nơi xuất phát, sao cho tổng chi phí là ít nhất.

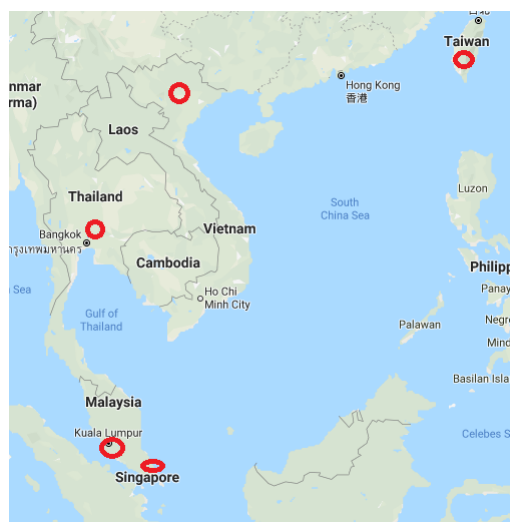


3

## Phương pháp tham lam: Bài toán người du lịch

Giả sử có các thành phố là: Hà Nội, Băng Cốc, Kualalumpur, Singapore, Đài Bắc.

**Có bao nhiêu cách để đi???**



4

## Phương pháp tham lam: Bài toán người du lịch

1. Hà Nội, Băng Cốc, Kualalumpur, Singapore, Đài Bắc, Hà Nội
2. Hà Nội, Băng Cốc, Kualalumpur, Đài Bắc, Singapore, Hà Nội
3. Hà Nội, Băng Cốc, Đài Bắc, Singapore,, Kualalumpur, Hà Nội
4. Hà Nội, Băng Cốc, Đài Bắc, Kualalumpur, Singapore, Hà Nội
5. Hà Nội, Băng Cốc, Singapore, Kualalumpur, Đài Bắc, Hà Nội
6. Hà Nội, Kualalumpur, Băng Cốc, Singapore, Đài Bắc, Hà Nội
7. Hà Nội, Kualalumpur, Băng Cốc, Đài Bắc, Singapore, Hà Nội
8. ....



5

## Phương pháp tham lam: Bài toán người du lịch

**Giả sử có 100  
thành phố thì  
có bao nhiêu  
cách???**



6

## Phương pháp tham lam: Bài toán người du lịch

**Giả sử có 100  
thành phố thì có  
bao nhiêu cách???**

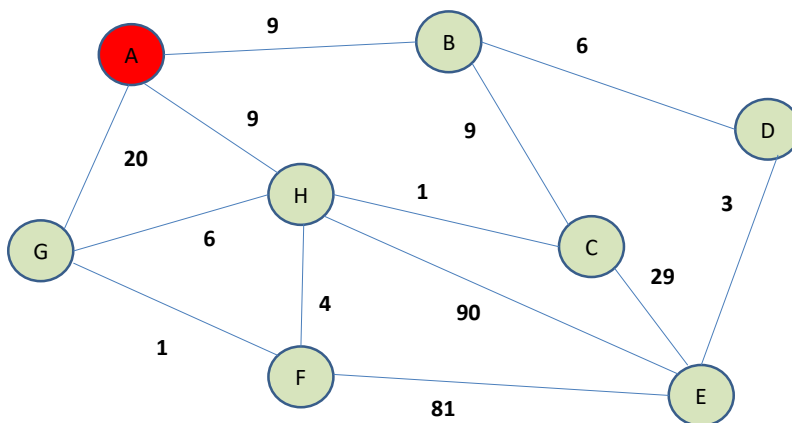
→ 99! cách

**=9.332622e+155**



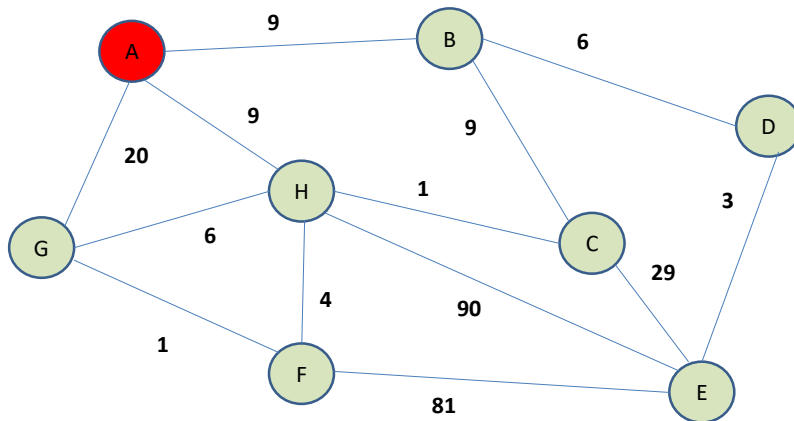
7

## Phương pháp tham lam: Bài toán người du lịch



8

## Phương pháp tham lam: Bài toán người du lịch



Phương pháp vét cạn:

Liệt kê tất cả các chu trình xuất phát từ A sau đó kiểm tra.

Độ phức tạp cho cách này là  $(n-1)!$  Nếu  $n$  lớn thì không giải quyết được.

Giải pháp:

Dùng phương pháp tham lam

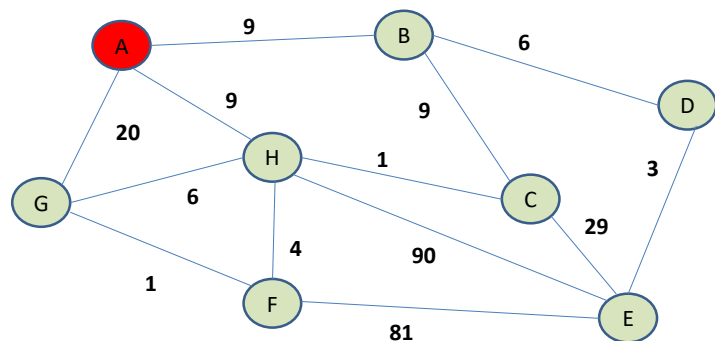
9

## Phương pháp tham lam: Bài toán người du lịch

Chọn đường đi có chi phí nhỏ nhất để đi

$A \rightarrow B \rightarrow D \rightarrow E \rightarrow C \rightarrow H \rightarrow F \rightarrow G \rightarrow A$

Độ phức tạp là  $O(n^2)$



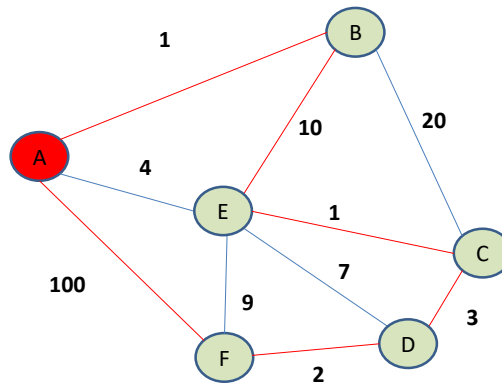
10

## Phương pháp tham lam: Bài toán người du lịch

Chọn đường đi có chi phí nhỏ nhất để đi!!!

→ Giải pháp không tối ưu

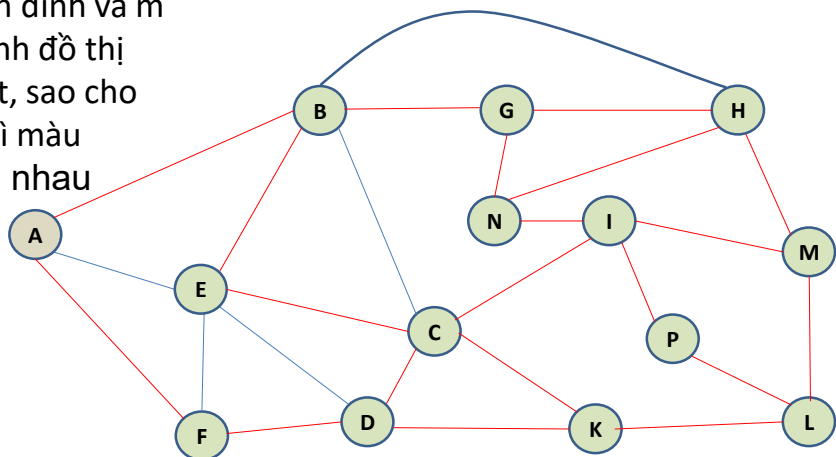
**Cho biết trường hợp tối ưu cho bài toán này???**



11

## Phương pháp tham lam : Tô màu đồ thị

Cho đồ thị gồm có  $n$  đỉnh và  $m$  cạnh. Hãy tô các đỉnh đồ thị bằng số màu ít nhất, sao cho hai đỉnh kề nhau thì màu không được giống nhau



12

## Phương pháp tham lam: Tô màu đồ thị

### Basic Greedy Coloring Algorithm:

1. Color first vertex with first color.
2. Do following for remaining  $V-1$  vertices.
  - a) Consider the currently picked vertex and color it with the lowest numbered color that has not been used on any previously colored vertices adjacent to it. If all previously used colors appear on vertices adjacent to  $v$ , assign a new color to it.

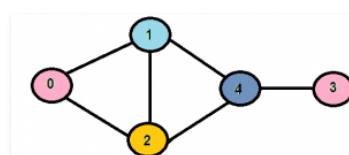
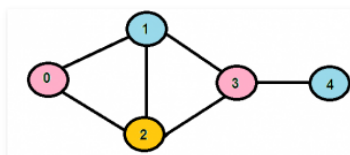
13

## Phương pháp tham lam: Tô màu đồ thị

Time Complexity:  $O(V^2 + E)$  in worst case.

### Analysis of Basic Algorithm

The above algorithm doesn't always use minimum number of colors. Also, the number of colors used sometime depend on the order in which vertices are processed. For example, consider the following two graphs. Note that in graph on right side, vertices 3 and 4 are swapped. If we consider the vertices 0, 1, 2, 3, 4 in left graph, we can color the graph using 3 colors. But if we consider the vertices 0, 1, 2, 3, 4 in right graph, we need 4 colors.



So the order in which the vertices are picked is important. Many people have suggested different ways to find an ordering that work better than the basic algorithm on average. The most common is [Welsh-Powell Algorithm](#) which considers vertices in descending order of degrees.

14

## Phương pháp tham lam: Tô màu đồ thị

---

### VERTEX COLORING: WELSH-POWELL ALGORITHM

One algorithm that gives a good solution to a vertex-coloring problem is the *Welsh-Powell algorithm*. It may not always give the best solution, but it will usually perform better than just coloring the vertices without a plan will.

The Welsh-Powell algorithm consists of the following steps:

1. Find the valence for each vertex.
2. List the vertices in order of descending valence (you can break ties any way you wish).
3. Color the first vertex in the list (the vertex with the highest valence) with color 1.
4. Go down the list and color every vertex not connected to the colored vertices above the same color. Then cross out all colored vertices in the list.
5. Repeat the process on the uncolored vertices with a new color – always working in descending order of valence until all the vertices have been colored.

15

## Phương pháp tham lam: Tô màu đồ thị

---

Thuật toán

1. Sắp xếp các đỉnh theo bậc giảm dần
2. Mau=0;
3. Trong khi các đỉnh chưa được tô màu hết
  - mau++;
  - Chọn đỉnh chưa được tô màu trong danh sách đỉnh theo thứ tự bậc từ cao đến thấp.
  - Tô bằng màu hiện tại cho các đỉnh chưa được tô màu sao cho các đỉnh được chọn tô không kề nhau.

16



## Phương pháp tham lam: Tô màu đồ thị

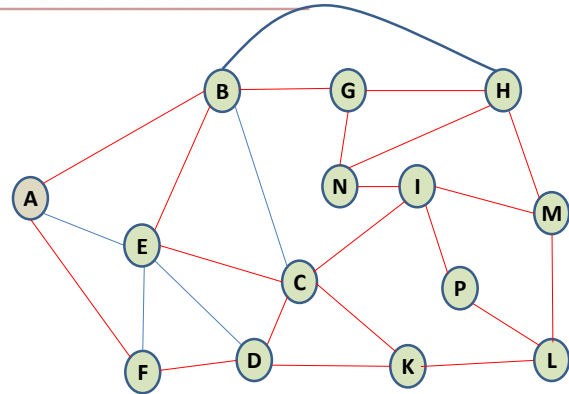
Tô màu đồ thị

Bậc đỉnh A= 3

Bậc đỉnh B= 5

Bậc đỉnh C= 5

-----



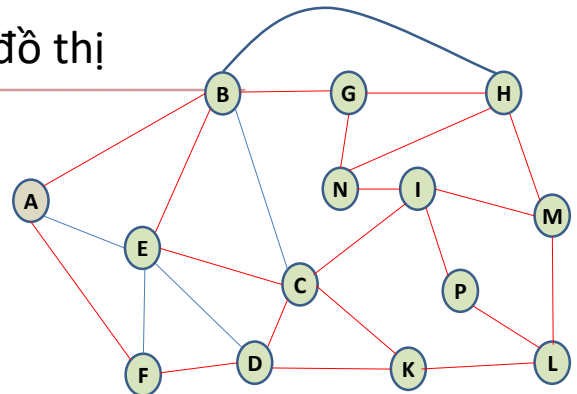
A	B	C	D	E	F	G	H	I	K	L	M	N	P
3	5	5	4	5	3	3	4	4	3	3	3	3	2

17

## Phương pháp tham lam: Tô màu đồ thị

Màu = 0

-----



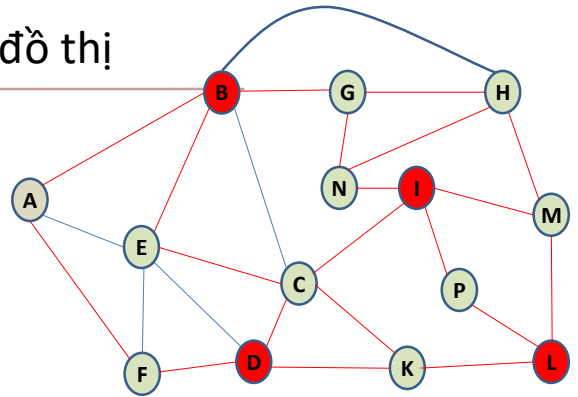
A	B	C	D	E	F	G	H	I	K	L	M	N	P
3	5	5	4	5	3	3	4	4	3	3	3	3	2

B	C	E	D	H	I	A	F	G	K	L	M	N	P
5	5	5	4	4	4	3	3	3	3	3	3	3	2

18

## Phương pháp tham lam: Tô màu đồ thị

Màu = 1



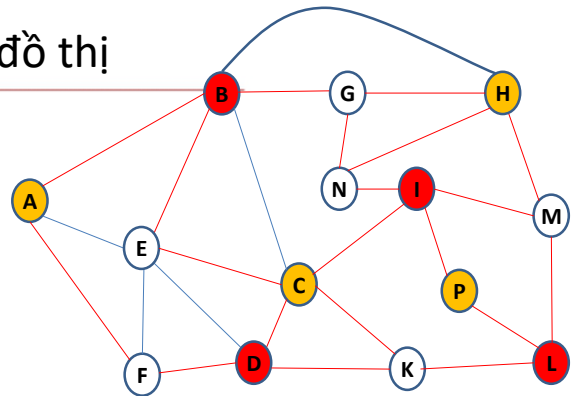
A	B	C	D	E	F	G	H	I	K	L	M	N	P
3	5	5	4	5	3	3	4	4	3	3	3	3	2

B	C	E	D	H	I	A	F	G	K	L	M	N	P
5	5	5	4	4	4	3	3	3	3	3	3	3	2

19

## Phương pháp tham lam: Tô màu đồ thị

Màu = 2



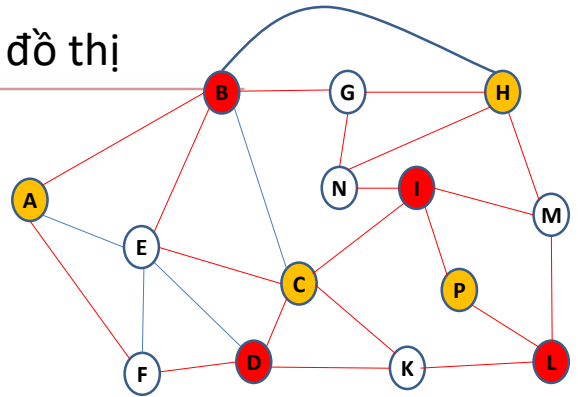
A	B	C	D	E	F	G	H	I	K	L	M	N	P
3	5	5	4	5	3	3	4	4	3	3	3	3	2

B	C	E	D	H	I	A	F	G	K	L	M	N	P
5	5	5	4	4	4	3	3	3	3	3	3	3	2

20

## Phương pháp tham lam: Tô màu đồ thị

Màu = 2



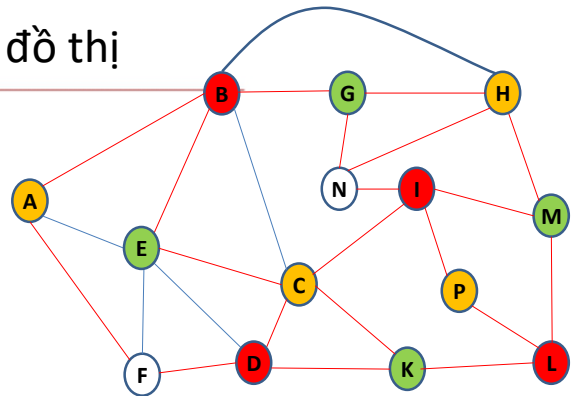
A	B	C	D	E	F	G	H	I	K	L	M	N	P
3	5	5	4	5	3	3	4	4	3	3	3	3	2

B	C	E	D	H	I	A	F	G	K	L	M	N	P
5	5	5	4	4	4	3	3	3	3	3	3	3	2

21

## Phương pháp tham lam: Tô màu đồ thị

Màu = 3



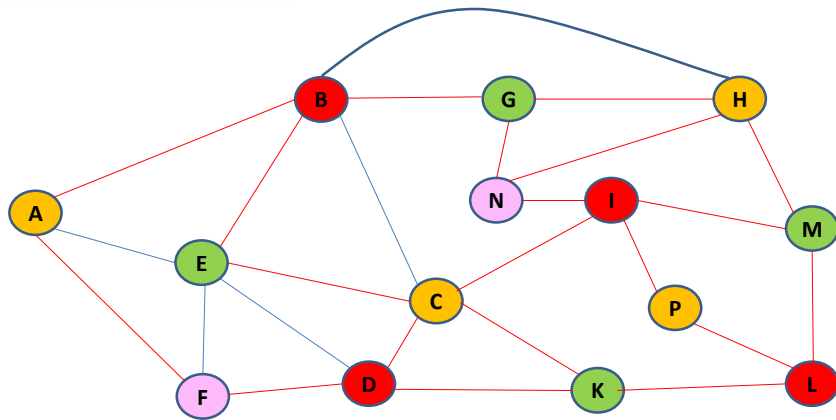
A	B	C	D	E	F	G	H	I	K	L	M	N	P
3	5	5	4	5	3	3	4	4	3	3	3	3	2

B	C	E	D	H	I	A	F	G	K	L	M	N	P
5	5	5	4	4	4	3	3	3	3	3	3	3	2

22

## Phương pháp tham lam: Tô màu đồ thị

Màu = 4

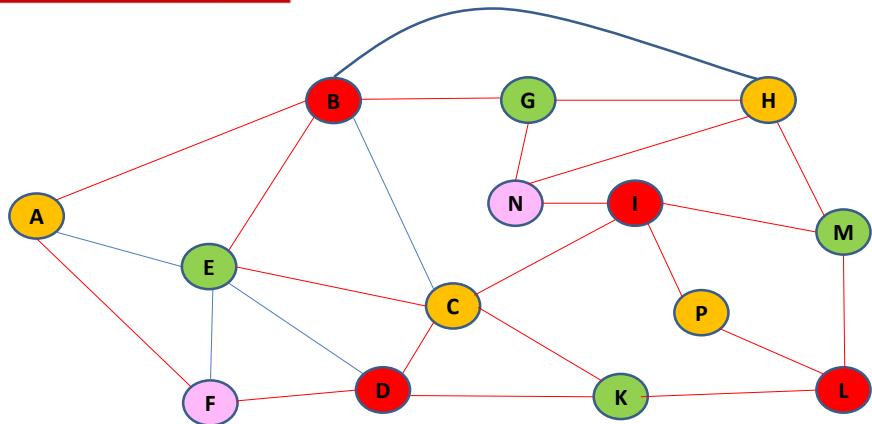


B	C	E	D	H	I	A	F	G	K	L	M	N	P
5	5	5	4	4	4	3	3	3	3	3	3	3	2

23

## Phương pháp tham lam: Tô màu đồ thị

Màu = 4



B	C	E	D	H	I	A	F	G	K	L	M	N	P
5	5	5	4	4	4	3	3	3	3	3	3	3	2

24

## Greedy Method Examples

---

### Problem 1

Given an array  $F$  with size  $n$ . Assume the array content  $F[i]$  indicates the length of the  $i_{th}$  file and we want to merge all these files into one single file. Check whether the following algorithm gives the best solution for this problem or not?

26-Feb-20

25

25

## Greedy Method Examples

---

### Problem 1

Given an array  $F$  with size  $n$ . Assume the array content  $F[i]$  indicates the length of the  $i_{th}$  file and we want to merge all these files into one single file. Check whether the following algorithm gives the best solution for this problem or not?

**Algorithm 1:** Merge the files contiguously. That means select the first two files and merge them. Then select the output of the previous merge and merge with the third file, and keep going...

26-Feb-20

26

26

## Greedy Method Examples

---

### Problem 1

Given an array  $F$  with size  $n$ . Assume the array content  $F[i]$  indicates the length of the  $i$ th file and we want to merge all these files into one single file. Check whether the following algorithm gives the best solution for this problem or not?

**Algorithm 2:** Merge the files in pairs. That means after the first step, the algorithm produces the  $n/2$  intermediate files. For the next step, we need to consider these intermediate files and merge them in pairs and keep going.

26-Feb-20

27

27

## Greedy Method Examples

---

### Problem 2

In Problem-1, what is the best way to merge all the files into a single file?

**Solution:** Using the Greedy algorithm we can reduce the total time for merging the given files.

26-Feb-20

28

28

## Greedy Method Examples

---

### Problem 2

In Problem-1, what is the best way to merge all the files into a single file?

**Solution:** Using the Greedy algorithm we can reduce the total time for merging the given files.

**Algorithm:**

1. Store file sizes in a priority queue. The key of elements are file lengths.
2. Repeat the following until there is only one file:
  - a. Extract two smallest elements X and Y.
  - b. Merge X and Y and insert this new file in the priority queue.

26-Feb-20

29

29

## Greedy Method Examples

---

### Problem 2

The given array is:  $F = \{10, 5, 100, 50, 20, 15\}$

After sorting  $\{5, 10, 15, 20, 50, 100\}$ .

**The total cost of merging = ???**

26-Feb-20

30

30

## Greedy Method Examples

---

### Problem 2

Ex: The given array is:  $F = \{10, 5, 100, 50, 20, 15\}$

After sorting  $\{5, 10, 15, 20, 50, 100\}$ .

We need to merge the two smallest files (5 and 10 size files) and as a result we get the following list of files:  $\{15, 15, 20, 50, 100\}$

Similarly  $\rightarrow \{20, 30, 50, 100\} \rightarrow \{50, 50, 100\} \rightarrow \{100, 100\} \rightarrow \{200\}$

The total cost of merging = Cost of all merging operations =  $15 + 30 + 50 + 100 + 200 = 395$ .

26-Feb-20

31

31

## Bài tập về nhà

---

Viết chương trình trộn  $n$  file số nguyên đã được sắp xếp tăng dần thành một file theo định dạng kiểu tham số dòng lệnh:

**merge input1 input2 ... inputn output**

Ví dụ:

merge ip1.txt ip2.txt ip3.txt out.txt

26-Feb-20

32

32



## Bài tập nhóm 3

---

Cho biết số lần dịch chuyển viên bi ít nhất khi đảo chiều 2 loại bi trên khay.



33

## Bài tập nhóm 3

---

**Cho biết số lần dịch chuyển viên bi ít nhất trong các trường hợp dưới đây???**



34

## Bài tập nhóm 3

---

**Cho biết số lần dịch chuyển viên bi ít nhất trong các trường hợp dưới đây???**



35

## Bài tập nhóm 3

---

Nhập n, cho biết số lần di chuyển ít nhất là bao nhiêu? Minh họa các bước di chuyển.



36

## Tài liệu đọc thêm

---

- <https://brilliant.org/wiki/greedy-algorithm/>
- <https://www.guru99.com/greedy-algorithm.html>
- [https://www.researchgate.net/publication/315849808\\_The\\_Application\\_of\\_Greedy\\_Algorithm\\_in\\_Real\\_Life](https://www.researchgate.net/publication/315849808_The_Application_of_Greedy_Algorithm_in_Real_Life)

37

## Link YouTube

---

- [https://www.youtube.com/watch?v=ARvQcqJ\\_-NY](https://www.youtube.com/watch?v=ARvQcqJ_-NY)



- <https://www.youtube.com/watch?v=tKwnms5iRBU>



38