1. Tóm tắt đề bài:

* Input:
  + Hai số nguyên dương ***n, m***: số lượng phần từ của 2 mảng ***n[], m[].***
  + Các số nguyên ***,***  : lần lượt là các phần tử của 2 mảng ***n[], m[].***
* Output:
  + Các phần tử của 2 mảng gộp lại được sắp xếp tăng dần.

1. Solution

* Subtask 1: đối với số lượng phần tử không quá lớn ta có thể sử dụng cách gộp 2 mảng thành 1 mảng mới rồi sắp xếp lại mảng mới.
* Subtask 2:
* Khi mảng có nhiều phần tử thì cách gộp 2 mảng rồi sắp xếp không khả thi vì độ phức tạp của thuật toán cao đồng thời đã bỏ qua chi tiết 2 mảng ban đầu đã được sắp xếp.
* Với mảng ***n[]*** được sắp xếp tăng dần và mảng được ***m[]*** sắp xếp giảm dần, bằng cách so sánh phần tử nhỏ nhất chưa được chọn ở mảng ***n[]*** và phần tử nhỏ nhất chưa được chọn ở mảng ***m[]***, phần tử nhỏ hơn sẽ được chọn vào mảng đích.
* Giải pháp: sử dụng kĩ thuật 2 con trỏ ([Xem thêm](https://vnoi.info/wiki/algo/basic/two-pointers.md))
* Mảng ***n[]*** có con trỏ i, con trỏ này bắt đầu ở vị trí đầu mảng ***n[]***.
* Con trỏ i này được thể hiện như phần tử nhỏ nhất chưa được chọn trong mảng ***n[]***, do mảng được sắp xếp tăng dần
* Mảng ***m[]*** có con trỏ j, con trỏ này bắt đầu ở vị trí cuối mảng ***m[]*** .
* Con trỏ j này được thể hiện như phần tử nhỏ nhất chưa được chọn trong mảng ***m[]***, do mảng được sắp xếp giảm dần
* Ta sẽ lặp lại công việc này, cho đến khi đưa hết các phần tử trong mảng ***n[]*** và ***m[]*** vào mảng đích
* Khi các phần tử trong 1 trong 2 mảng đều đã được đưa vào mảng đích thì đưa lần lượt các phần tử trong mảng còn lại vào mảng đích.
* Ngược lại:
* So sánh hai phần tử ở hai con trỏ.
* Đưa phần tử có giá trị nhỏ hơn vào dãy c, nếu hai phần tử có giá trị như nhau thì chọn một trong hai.
* Tăng vị trí con trỏ ở phần tử được đưa vào lên một đơn vị.