

BÁO CÁO BÀI GIỮA KỲ

Bài 4

1. Mô tả cách giải quyết.

Dữ liệu được lấy trực tiếp từ file Excel, mỗi phần tử lấy ra được thêm và đồng thời sắp xếp(giảm dần) mỗi phần tử đó vào 1 cấu trúc dữ liệu, bộ dữ liệu trả về sau khi thêm đã được sắp xếp, việc còn lại chỉ là in dữ liệu ra theo điều kiện đề bài. 2 hàm in, in ra toàn bộ danh sách và in ra số lượng phần tử nhất định. Cách giải quyết cụ thể như sau:

- Cấu trúc là danh sách liên kết có các Node gồm 3 thành phần dữ liệu lưu tên tỉnh, giá trị, và con trỏ.
- Danh sách liên kết được khởi tạo với 3 thành phần để kiểm soát phần tử đầu, cuối và độ dài danh sách.
- Sử dụng thuật toán chính là thêm và sắp xếp (giảm dần):
 - + Node đầu tiên sẽ được thêm vào head danh sách(head và tail trùng nhau).
 - + Node thứ 2 nhỏ hơn Node đầu thì: thêm vào sau tail và tail được cập nhật lại là Node này, ngược lại ta thêm vào trước head và head được cập nhật lại là Node này.
 - + Từ Node thứ 3 trở đi cũng được xét 2 trường hợp trên và nếu không thuộc 2 trường hợp đó thì sẽ duyệt lần lượt 2 Node liên nhau đôi một, nếu giá trị Node đó thuộc khoảng 2 giá trị 2 Node liên nhau thì thêm vào.

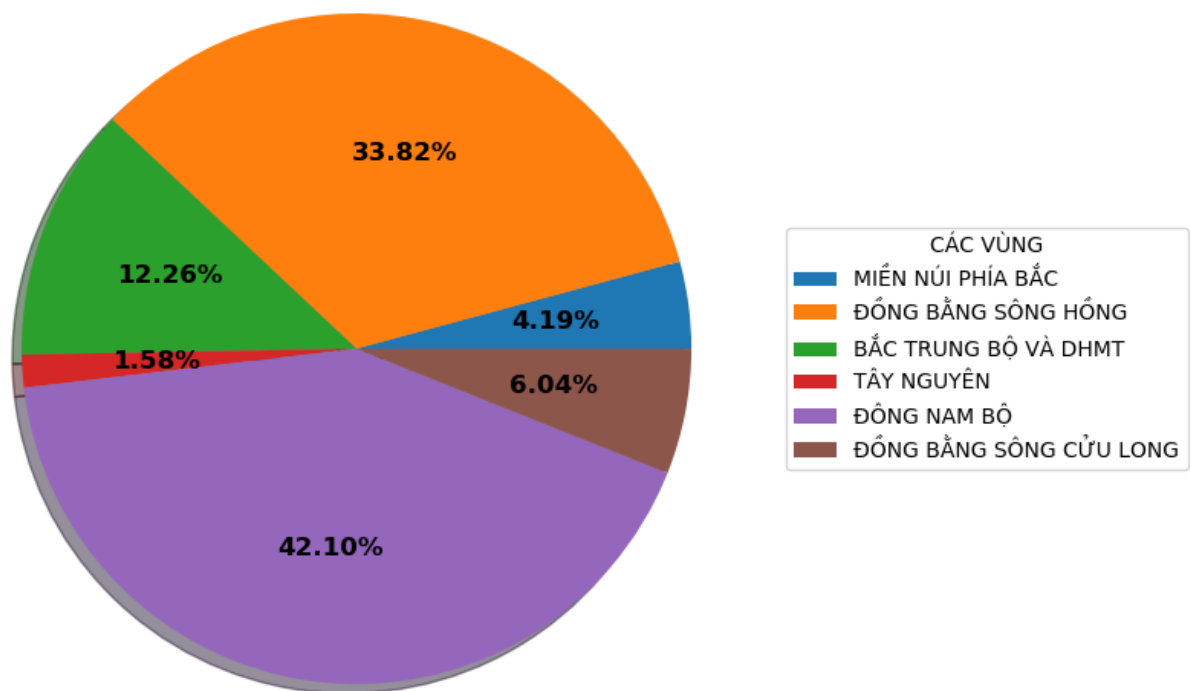
2. Sử dụng cấu trúc dữ liệu nào, tại sao lại chọn cấu trúc đó.

Cấu trúc dữ liệu được sử dụng để lưu trữ dữ liệu là Danh sách liên kết đơn. Vì các liên kết tới phần tử khác tường minh nên với cấu trúc này các phần tử trong danh sách không cần phải lưu trữ liên tục trong bộ nhớ, cấu trúc này còn cho phép các thao tác thêm, xóa trên danh sách được thực hiện dễ dàng, phản ánh được bản chất linh động của danh sách so với mảng. Với bộ dữ liệu 6 vùng chứa 63 tỉnh/TP và các cột số liệu khác thì cấu trúc này đủ linh động để tổ chức các dữ liệu đó và xử lý phù hợp theo từng bài toán.

3. Mô tả thuật toán (giả code) xử lý chính.

```
Def AppendAndSort(Node)
  if head == None
    Node = head
    Node = tail
  if Node.data >= head.data
    Node.next = head
    head = Node
  else
    if Node.data <= tail.data
      tail.next = Node
      tail = Node
    if Node.data > tail.data
      cur = head
      while cur.data != tail
        if Node.data < cur.data and Node.data > cur.next.data
          Node.next = cur.next
          cur.next = Node
        cur = cur.next
```

4. Kết quả thu được: ảnh, biểu đồ.



2. Nhập vào số k, trả về k tỉnh có thu ngân sách cao nhất theo thứ tự giảm dần.

k = 3

STT	Tên Tỉnh	
1	TP. HỒ CHÍ MINH	399125000
2	HÀ NỘI	263111000
3	BÀ RỊA VŨNG TÀU	74397000

3. Nhập vào số k, trả về k tỉnh nhận được bổ sung từ ngân sách nhà nước nhiều nhất theo thứ tự giảm dần.

k = 3

STT	Tên Tỉnh	
1	THANH HÓA	15355034
2	NGHỆ AN	10798616
3	ĐẮK LẮK	7821082

4. Nhập vào số k, trả về k tỉnh có đóng góp vào ngân sách nhà nước cao nhất.

k = 3

STT	Tên Tỉnh	
1	TP. HỒ CHÍ MINH	324874486
2	HÀ NỘI	162569209
3	BÀ RỊA VŨNG TÀU	57233684

5. Nhập vào tên một vùng, trả về tên các tỉnh trong vùng cùng tổng thu ngân sách của tỉnh đó theo thứ tự giảm dần.

Nhập tên vùng(in hoa, có dấu): TÂY NGUYÊN

STT	Tên Tỉnh	Thu Ngân Sách(TrĐ)
1	LÂM ĐỒNG	7695000
2	ĐẮK LẮK	5442500
3	GIA LAI	4503600
4	KON TUM	2399700
5	ĐẮK NÔNG	2358500

6. Liệt kê tên các tỉnh (ít nhất) có tổng nguồn thu ngân sách chiếm tới 50% tổng thu của cả nước.

STT	Tên Tỉnh	Thu Ngân Sách(TrĐ)
1	TP. HỒ CHÍ MINH	399125000
2	HÀ NỘI	263111000
3	QUẢNG NINH	39091000
4	VĨNH LONG	6170200

5. Đánh giá của cá nhân về thuật toán, về cấu trúc dữ liệu, về số liệu, về đặc điểm của số liệu, về kết quả.

Về thuật toán:

- Thuật toán sắp xếp có độ phức tạp $O(n)$ coi là ổn so với các thuật toán sắp xếp khác (thường có độ phức tạp $O(n \log n)$, $O(n^2)$).
- Khác với các thuật toán sắp xếp khác là sắp xếp trên bộ dữ liệu có sẵn thì thuật toán này kết hợp với các hàm (thêm vào đầu, vào cuối, vào sau một phần tử) để sắp xếp ngay khi phần tử đó được thêm vào danh sách.
- Điểm yếu của thuật toán này có thể thấy rõ là với bộ dữ liệu đã có sẵn thì không thể sắp xếp được.

Về cấu trúc dữ liệu: So với bài assignment cài đặt danh sách móc nối của em làm trước đó thì ADT này được cải tiến thêm con trỏ tail kiểm soát Node cuối danh sách giúp phương thức thêm 1 Node vào cuối danh sách có độ phức tạp từ $O(n)$ thành $O(1)$.

Về số liệu, đặc điểm số liệu: Bộ dữ liệu phù hợp với thuật toán và cấu trúc dữ liệu, nhưng với bộ dữ liệu lớn thì việc lấy ra, và xử lý toàn bộ dữ liệu và trích xuất dữ liệu theo yêu cầu người dùng của thuật toán này lại không ổn vì khi người dùng muốn trích xuất 1 lượng nhỏ dữ liệu mà thuật toán phải xử lý cả bộ dữ liệu thì sẽ tốn kém thời gian cho người sử dụng, vì thế có 1 số thuật toán và cấu trúc dữ liệu khác hiệu quả bằng hoặc hơn thuật toán này.

Về kết quả:

- Bảng kết quả ở trên rất là rõ ràng, in ra số thứ tự các tỉnh cũng như các số liệu một cách trực quan.
- Về bài làm thứ 6, được sử dụng khi bộ dữ liệu đã sắp xếp, chương trình được viết 1 cách chủ quan, có thể bộ dữ liệu khác trả về kết quả sai. Vì chưa kiểm nghiệm tính đúng đắn, cũng như không
- Nhìn chung thì tập kết quả trả về là đúng so với yêu cầu bài toán.