

# ĐẶT VẤN ĐỀ

## ❖ QUẢN LÝ VÙNG NHỚ

Vùng nhớ chương trình được quản lý theo hai cách:

- Quản lý tự động: Được thực hiện khi khai báo biến, hằng trong C/C++. Các đối tượng này được gọi là biến cấp phát tĩnh (biến tĩnh, biến nửa tĩnh).
- Quản lý do người lập trình: Được thực hiện khi cấp phát vùng nhớ. Các vùng nhớ được cấp phát được gọi là biến cấp phát động (biến động)

# ĐẶT VẤN ĐỀ

## ❖ BIẾN CẤP PHÁT TĨNH

Là các biến được tạo bằng khai báo. Ví dụ:

```
struct PS{  
    int ts, ms;  
};  
int x, y;  
char s[50];  
PS p;
```

# ĐẶT VẤN ĐỀ

## ❖ BIẾN CẤP PHÁT TĨNH

Biến cấp phát tĩnh (biến tĩnh, biến nửa tĩnh) có các đặc điểm sau:

- Được khai báo tường minh, có tên gọi.
- Tồn tại trong phạm vi khai báo.
- Được cấp phát trong stack segment hoặc data segment.
- Kích thước không đổi. Ví dụ: kích thước mảng.

# ĐẶT VẤN ĐỀ

## ❖ BIẾN CẤP PHÁT ĐỘNG

Là các biến được tạo khi cấp phát vùng nhớ. Ví dụ:

```
struct PS{  
    int ts, ms;  
};
```

- Tạo vùng nhớ cho biến kiểu nguyên:

`malloc(sizeof(int));` hoặc `new int;`

- Tạo vùng nhớ cho biến kiểu PS:

`malloc(sizeof(PS));` hoặc `new PS;`

# ĐẶT VẤN ĐỀ

## ❖ BIẾN CẤP PHÁT ĐỘNG

Biến cấp phát động (biến động) có các đặc điểm sau:

- Không được khai báo tường minh, không có tên gọi.
- Cấp phát khi cần sử dụng, khi hết sử dụng phải giải phóng.
- Được cấp phát trong heap segment.
- Kích thước vùng nhớ được cấp phát tùy theo yêu cầu và giới hạn bộ nhớ.

# ĐẶT VẤN ĐỀ

## ❖ NHƯỢC ĐIỂM CỦA CẤP PHÁT TÍNH

- Lãng phí bộ nhớ.

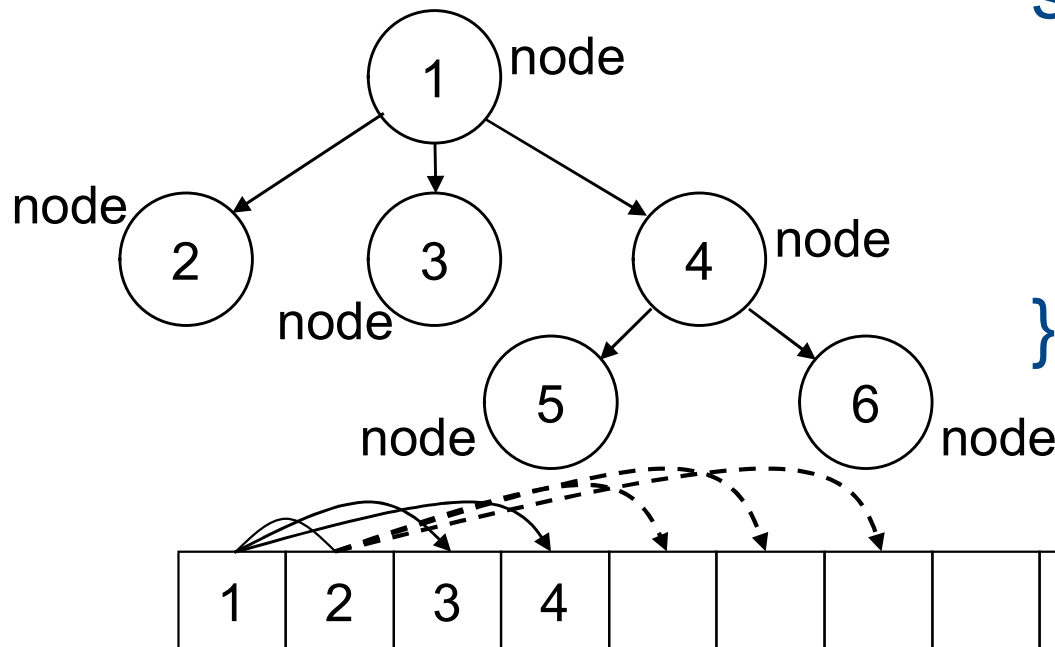
Ví dụ: danh sách nhân viên của một công ty có thể thay đổi.

Giải pháp: xác định kích thước tối đa  $n$  của danh sách, khai báo mảng với kích thước  $n \Rightarrow$  lãng phí bộ nhớ.

# ĐẶT VẤN ĐỀ

## ❖ NHƯỢC ĐIỂM CỦA CẤP PHÁT TĨNH

- Không thể định nghĩa kiểu có cấu trúc đệ quy vì không xác định được kích thước của nó. Ví dụ: tổ chức dữ liệu cho cây:



```
struct Node {  
    int Key;  
    Node left, mid, right;  
};
```