



## B12-Cay huffman - hhhh

Hệ điều hành (Trường Đại học Công nghệ thông tin, Đại học Quốc gia Thành phố Hồ Chí Minh)

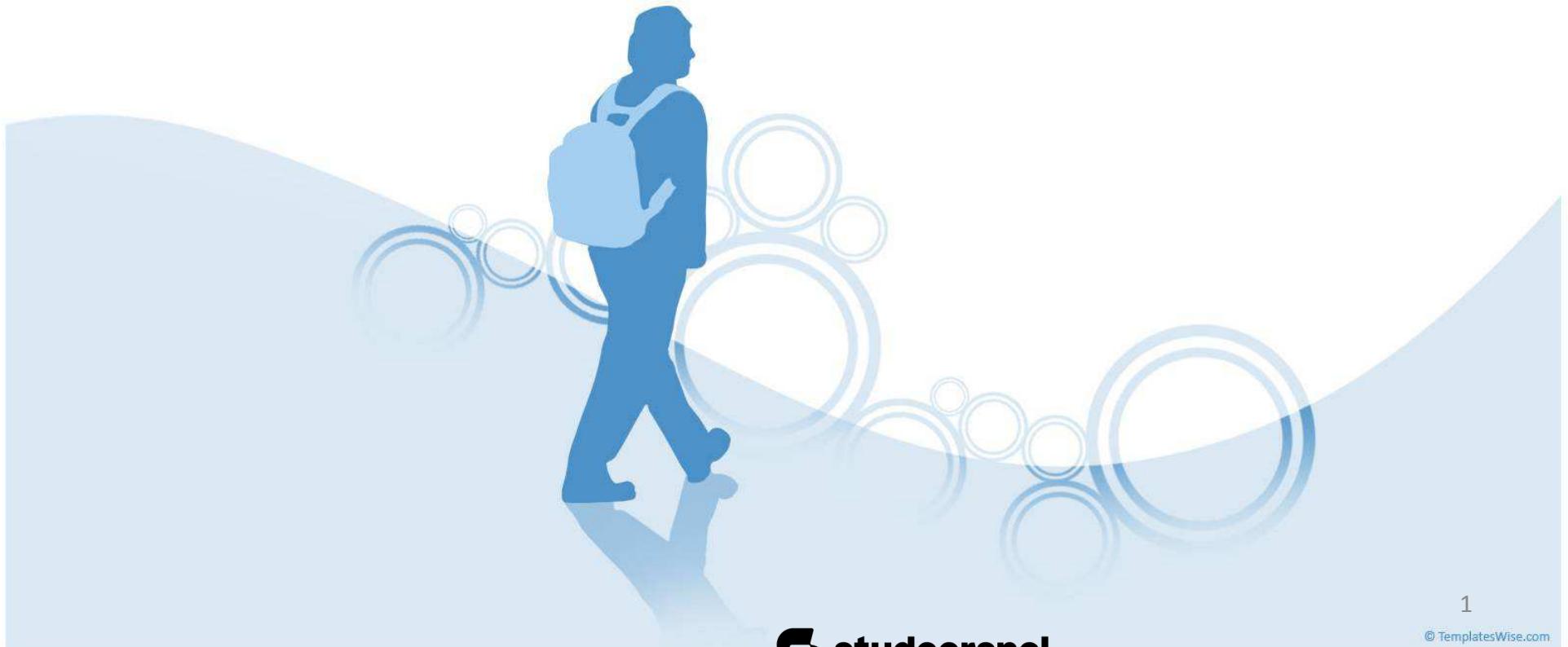


Scan to open on Studeersnel

# Giới thiệu



## Cây huffman



1

# Nội dung trình bày



- Mô hình mã hóa huffman
  - Cây huffman
  - Xác định mã, mã hóa, giải mã
- Triển cây huffman trên mảng

# Nén với mã huffman



- Nén huffman không mất thông tin
  - Nén không mất thông tin
  - Dựa trên thống kê tần số: những ký tự xuất hiện nhiều lần, sử dụng ký mã mới ít bit hơn
  - Thuật toán hiệu quả sử dụng: nén jpg, ...

# Nén với mã huffman



- Ý tưởng
  - Xây dựng bộ mã huffman
  - Đọc ký tự, chuyển sang dãy bit mới
  - Gửi đến nơi nhận
  - Đọc dãy bit, nhận dạng ký tự chuyển về ký tự cũ

# Xây dựng mã huffman



- **Ý tưởng**
  - Xây dựng bảng thống kê tần số xuất hiện của các ký tự cần mã hóa
  - Mỗi phần tử được xem như là đỉnh của một cây
  - Lắp cho đến lúc chỉ còn một cây
    - Chọn 2 cây có trọng số bé nhất ghép thành một cây mới
  - Từ đỉnh duyệt cây
    - Nếu về bên trái chọn bit 0
    - Về phải chọn bit 1
    - Đến lá thì dãy bit đã duyệt chính là mã mới của ký tự

# Xây dựng mã huffman



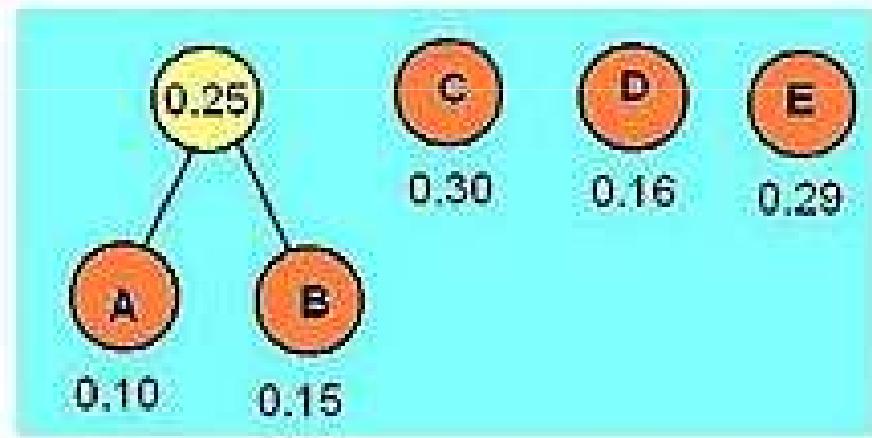
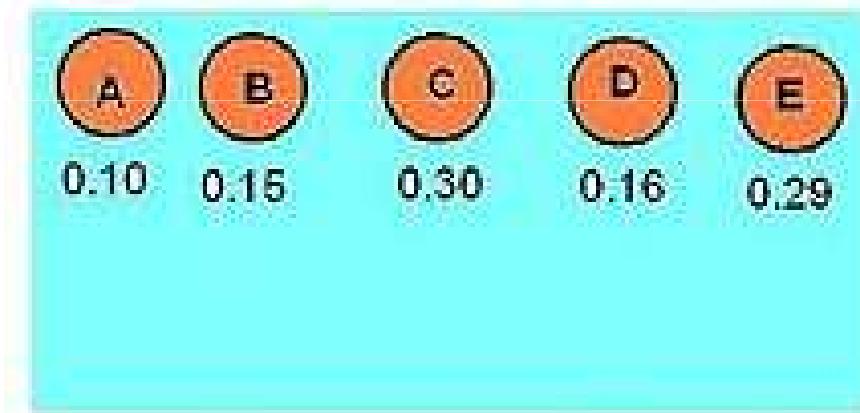
- Ví dụ

A	B	C	D	E
0.10	0.15	0.30	0.16	0.29

# Xây dựng mã huffman



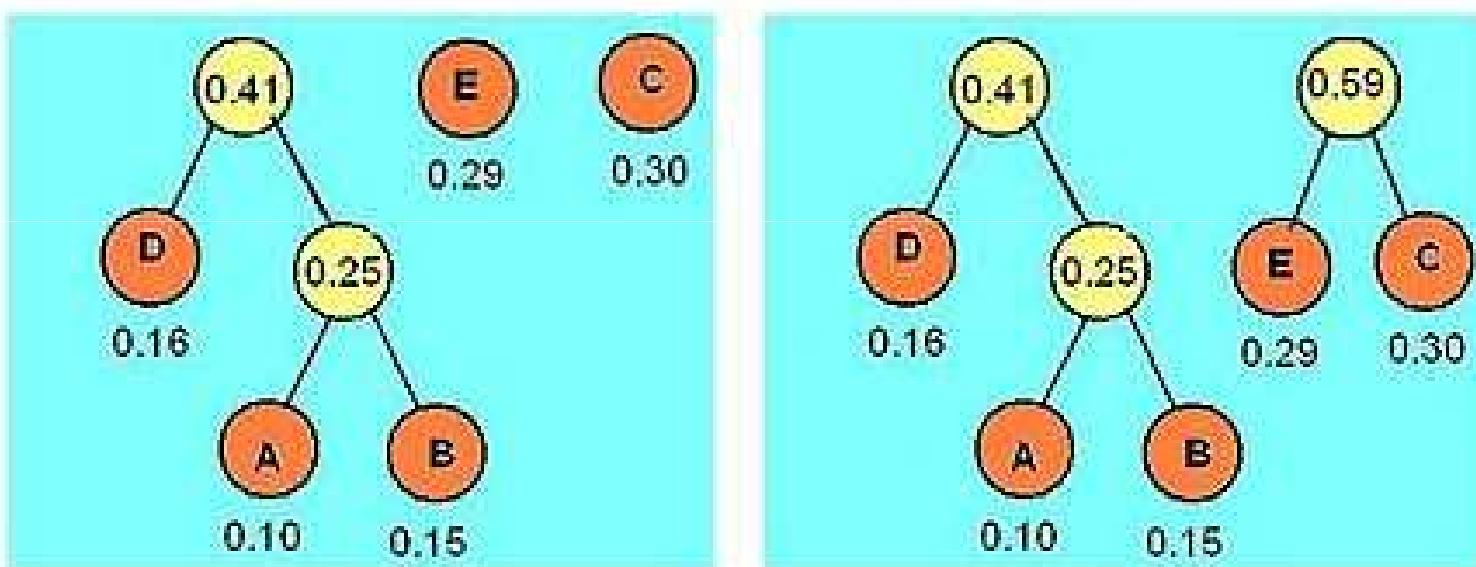
- Ví dụ



# Xây dựng mã huffman



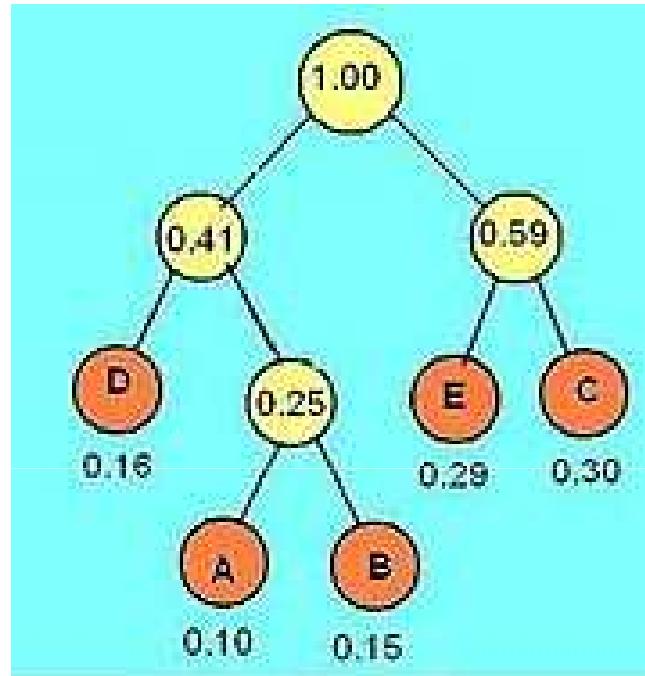
- Ví dụ



# Xây dựng mã huffman



- Ví dụ



A	B	C	D	E
010	011	11	00	10

# Triển khai thực tiễn



- Mô hình sử dụng cây
  - Xây dựng cấu trúc cây kiểu sử dụng con trỏ trái, phải
  - Sau đó sử dụng mảng lưu các cây (rừng)
  - Mỗi bước mảng giảm đi một phần tử cuối cùng sẽ có cây

# Triển khai thực tiễn



- Mô hình sử dụng mảng
  - Nếu có cây khi duyệt bit có thể sử dụng chiến lược duyệt từ duyệt từ lá lên đỉnh và đảo ngược xâu bit
  - Việc xây dựng cây có thể chỉ cần xây dựng từ lá, con chỉ lên cha của nó
  - Vì thế thay vì chọn một nút có hai con có thể thực hiện việc một con chỉ một link đến cha

# Triển khai thực tiễn



- Mô hình sử dụng mảng
  - Nếu có cây khi duyệt bit có thể sử dụng chiến lược duyệt từ duyệt từ lá lên đỉnh và đảo ngược xâu bit
  - Việc xây dựng cây có thể chỉ cần xây dựng từ lá, con chỉ lên cha của nó
  - Vì thế thay vì chọn một nút có hai con có thể thực hiện việc một con chỉ một link đến cha
  - Vì dùng mảng như là một nút
  - Mỗi bước thực hiện ghép hai đỉnh sẽ sinh một đỉnh mới vì thế cần  $2^*n-1$  đỉnh.

# Triển khai thực tiễn



- Mô hình sử dụng mảng (t)
  - Vì một đỉnh có thể bị dịch chuyển nên nếu lưu giá trị vào đỉnh sẽ không còn đúng về tham chiếu
  - Sẽ dùng một mảng để lưu đỉnh các cây mới
  - Dùng mảng chỉ số để lưu trữ các rừng

# Thuật toán



- Thuật toán tính cây huffman
  - Input: a[]: ký tự, b[]: tần suất, n
  - Output: p[]: mảng mô tả cây
  - For(i=0;i<2n;i++)
    - P[i]=i;//không quy định tự trả
  - For(i=0;i<n;i++)
    - T[i]=i;//
  - SortbybDesc(t,n);//sắp xếp chỉ số liên kết của t đến giá trị b[], giảm dần



- Thuật toán tính cây huffman (t)
  - C=n;//Số đỉnh trong rừng
  - Cc=n-1;//Số cây còn lại trong rừng
  - While(cc>0)
    - b[c]=b[t[cc]]+b[t[cc-1]]
    - P[t[cc]]=-c;//trái
    - P[t[cc-1]]=c;//phải
    - Cc--;
    - T[cc]=c;
    - C++;
    - Insert(t,cc);//thực hiện thuật toán cho đổi chỗ phần tử ở cuối vào vị trí thích hợp của nó, đảm bảo không tăng

# Thuật toán



- Thuật toán tính cây huffman (t)
  - For( $i=0; i < n; i++$ )
    - $Cr=i;$
    - $t = "$
    - Whiel( $P[cr] \neq cr$ )
      - If( $p[cr] < 0$ )
        - »  $T=t+'0'$
      - Else
        - »  $T=t+'1'$
      - $Cr=abs(p[cr])$
    - $Code[i]=invert(t); //đảo thứ tự các bit$

# Bài tập trên lớp



- Triển khai thuật toán với cây huffman

# Bài tập

- Triển khai thuật toán mã hóa và giải mã huffman

