Câu 4. Tên bài: **SHOES.CPP**

Theo bổ đề **Heuristic**: cho 2 dãy tăng dần các số *dương* $A_1, A_2, \ldots, A_n$ và $B_1, B_2, \ldots, B_n$. Gọi $C_1, C_2, \ldots, C_n$ là một hoán vị bất kì của dãy B. Khi đó:

$$|A_1 - B_1| + |A_2 - B_2| + \ldots + |A_n - B_n| \leq |A_1 - C_1| + |A_2 - C_2| + \ldots + |A_n - C_n|.$$

Như vậy, ta sẽ sắp xếp N đôi giày và K cầu thủ theo thứ tự tăng dần. Sau đó, ta lại đưa vào mảng 2 chiều *A[i][j]* tức là độ chênh lệch của cầu thủ i khi đi đôi giày j: *A[i][j] = | S[i] – H[j] |*

Sau đó làm tương tự câu 2 và 3 ở trên.

Độ phức tạp của thuật toán là *O(N * K)*.

Câu 5. Tên bài: **BUILT.CPP**

Ta thấy rằng, nếu kho A cung cấp x đơn vị vật liệu cho công trường thứ i thì đồng nghĩa với việc kho B phải cung cấp D[i] – x đơn vị vật liệu cho công trường đó.

Ta xây dựng hàm mục tiêu F(i, j) là cước phí vận chuyển nhỏ nhất trong trường hợp kho A cung cấp j đơn vị vật liệu đến các công trường từ 1, 2, .. i (j ≤ R). Công thức truy hồi :

+ F(1, j) = Min(A[1] * x + B[1] *(D[1] – x)) với x là đơn vị vật liệu kho A cung cấp (x ≤ D[1] và x ≤ j)

+ Do đó F(i, j) = Min {A[i]*x + B[i]*(D[i]-x) + F(i-1, j-x)}.

Kết quả bài toán là F[N[R]

Độ phức tạp của thuật toán là O(N * R * max(D[i])).

Câu 6. Tên bài: **CABLE.CPP**

+ Gọi F[i][0] là độ dài ngắn nhất xét đến máy i, với máy i không được nối với máy i-1. Gọi F[i][1] là độ dài ngắn nhất xét đến máy i, với máy i được nối với máy i-1

Khi đó: F[2][0] = +oo (không tồn tại cách này)

F[2][1] = a[1].

Lập công thức:

+ F[i][0] = F[i-1][1] (nếu i không nối i-1 thì bắt buộc i-1 phải nối i-2)

+ F[i][1] = min(F[i-1][0] , F[i-1][1]) + a[i-1] (i nối i-1 thì i-1 có thể nối hoặc không nối với i-2)

Kết quả bài toán là F[N][1]

Độ phức tạp của thuật toán là *O(N)*.