Thuật toán:

Trước hết, ta sort lại các cột sắt theo chiều tọa độ x tăng dần.

Sau đó, duyệt cột đầu tiên được chọn (giả sử là điểm R(x0, y0)).

Khi đó, tất cả các cây đều phải có tọa độ x >= x0, nếu không thì sẽ không thể bao hết được.

Xét một đường bao bất kì chứa được tất cả các cây:

Thứ nhất, đường bao đó phải là một đa giác lồi với các đỉnh là các cột sắt, trong đó cột ở (x0, y0) chắc chắn được chọn.

Thứ hai, giả sử ta có các cạnh của đa giác lồi theo chiều ngược kim đồng hồ. Vì các cây đều nằm trong đa giác nên mỗi cây đều nhìn các cạnh của đa giác theo chiều ngược kim đồng hồ. Nói cách khác, nếu cây đang xét ở điểm O, thì với mỗi cạnh AB của đa giác, ta có OA x OB > 0 (OA và OB ở đây là 2 vector).

Như vậy, ta có thể tính cách dựng đa giác tối ưu như sau:

Sort các cột sắt theo thứ tự ngược chiều kim đồng hồ so với điểm R(x0, y0). Nói cách khác, điểm A đứng trước điểm B nếu RA x RB > 0. Lưu chúng vào mảng A, sau đó thêm điểm R vào cuối mảng A.

Quy hoạch động dp[i] có nghĩa là chi phí tối thiểu để xây một bao lồi sao cho:

+ Bao lồi bắt đầu ở R, kết thúc ở điểm A[i] (bao lồi bị “hở”, nếu A[i] = R thì bao lồi kín và đây chính là đáp án).

+ Mọi cây đều nhìn các cạnh của bao lồi theo chiều ngược kim đồng hồ.

Ta xét điểm A[j], sao cho j < i. Nếu tất cả các cây đều nhìn cạnh (A[j],A[i]) theo chiều ngược kim đồng hồ (có nghĩa là OA[j] x OA[i] > 0 với mọi cây O) thì ta có thể tính dp[i] = min(dp[i], dp[j] + dist(A[i], A[j])) trong đó dist(A, B) là khoảng cách giữa 2 điểm A và B.

Đáp án chính là phần tử cuối cùng của mảng dp (khi A[i] = R).