

# Solution Lắp đặt cảm biến

Nhóm: 7

Tên thành viên:

- Vũ Minh Phương
- Võ Lê Ngọc Thịnh

## Nhận xét bài toán

Bài toán yêu cầu chọn một tập con các cảm biến sao cho toàn bộ lưới kích thước  $n \times n$  được bao phủ, đồng thời tổng chi phí là nhỏ nhất. Mỗi cảm biến có vị trí  $(x_i, y_i)$ , chi phí  $c_i$ , và phủ tất cả các ô nằm trong phạm vi Chebyshev bán kính  $R$ . Đây là một biến thể trực tiếp của bài toán **Minimum Set Cover**, một bài toán NP-hard. Do đó, không thể tìm lời giải tối ưu trong thời gian đa thức, và ta buộc phải dùng các kỹ thuật xấp xỉ và heuristic.

Giải pháp được xây dựng dựa trên bốn ý tưởng chính:

1. Biểu diễn vùng phủ của từng cảm biến bằng bitmask để tăng tốc độ hợp và đếm số ô mới được phủ.
2. Áp dụng chiến lược tham lam với tiêu chí chọn cảm biến có tỉ lệ  $\frac{\text{cost}}{\text{newCoverage}}$  nhỏ nhất.
3. Kết hợp random hóa để tránh rơi vào nghiệm tham lam kém.
4. Áp dụng bước tinh chỉnh cuối cùng (pruning) để loại bỏ các cảm biến thừa nhưng không làm mất độ bao phủ.

## Chứng minh tính đúng đắn

1. Mỗi bước greedy đảm bảo chọn được một cảm biến phủ ít nhất một ô chưa được phủ, nên số ô chưa được phủ giảm dần và thuật toán luôn kết thúc khi bao phủ toàn bộ lưới.
2. Nếu greedy rơi vào nghiệm cục bộ kém, random hóa tạo ra các cấu hình khởi tạo khác nhau giúp vượt qua điểm kẹt. Lặp lại 1000 lần làm tăng xác suất thu được nghiệm tốt.
3. Bước pruning đảm bảo tập cảm biến đầu ra là một tập rút gọn: nếu loại bỏ một cảm biến mà các cảm biến khác vẫn bao phủ toàn bộ lưới, cảm biến đó là không cần thiết.
4. Việc dùng bitmask để mô tả vùng phủ là chính xác tuyệt đối, vì mỗi ô trên lưới tương ứng với đúng một bit trong bitmask, và các phép toán hợp/hiệu được tính chính xác bằng toán tử bitwise.

## Phân tích độ phức tạp thời gian

- Mỗi cảm biến có thể phủ tối đa  $(2R + 1)^2$  ô nên bước tiền xử lý có độ phức tạp  $O(mR^2)$ .
- Trong mỗi vòng greedy, cần đếm số ô mới được phủ bằng phép bit\_count trên số nguyên lớn có kích thước  $n^2$  bit, nên mỗi vòng có độ phức tạp  $O(m \cdot n^2)$ .
- Thuật toán lặp lại 1000 lần thử ngẫu nhiên và chọn nghiệm tốt nhất, nên tổng thời gian xấp xỉ  $O(1000 \cdot m \cdot n^2)$ .
- Bước pruning kiểm tra xem cảm biến nào có thể bỏ được, có độ phức tạp xấp xỉ  $O(m^2 \cdot n^2)$  nhưng thực tế nhỏ hơn nhiều vì số cảm biến được chọn thường ít.

Với giới hạn  $n \leq 50$  và  $m \leq 100$ , thuật toán chạy nhanh và vượt qua toàn bộ bộ test.

## Phân tích độ phức tạp không gian

- Mỗi bitmask có kích thước  $n^2$  bit, tương đương khoảng  $n^2/8$  bytes.
- Có tổng cộng  $m$  cảm biến, nên tổng dung lượng lưu bitmask là  $O(m \cdot n^2)$ .
- Ngoài ra, các cấu trúc khác như danh sách cảm biến được chọn, danh sách tạm thời trong quá trình greedy và pruning đều có kích thước tối đa  $O(m)$ .

Suy ra, tổng độ phức tạp không gian của thuật toán là  $O(m \cdot n^2)$ .