**Bài 1: class**

Sub2 :n<=100000

Σ(i) Σ(j) (ci+cj) \* min(|ai-aj|,|bi-bj|) = Σ(i) ci \* Σ(j) \* min(|ai-aj|,|bi-bj|).

Như vậy ta chỉ cần tính tổng trên cho mỗi i rồi cộng tổng vào kết quả, với mỗi i ta xét 4 trường hợp của j:

* Ai > aj và bi > bj.
* Ai > aj và bi <= bj.
* Ai <= aj và bi > bj.
* Ai <= aj và bi <= bj.

Dùng cấu trúc dữ liệu để tính các trường hợp trên.

Đpt : O(N\*log2(N)^2).

Sub3:n<=500000,ci=cj

Do ci=cj với mọi i,j nên ta chỉ cần tính Σ(i) Σ(j) min(|ai-aj|,|bi-bj|).

Với mỗi i đặt x=ai-bi,y=ai+bi;

Ta có max(|ai-aj|,|bi-bj|) = (|xi-xj|+|yi-yj|)/2.

Do đó min(|ai-aj|,|bi-bj|) = |ai-aj| + |bi-bj| - max(|ai-aj|,|bi-bj|) = |ai-aj| + |bi-bj| - (|xi-xj|+|yi-yj|)/2.

Đến đây thì ta tính riêng tổng từng phần |ai-aj|, |bi-bj|, |xi-xj| , |yi-yj|, rồi gộp lại là đc kết quả.

Cách tính từng phần : VD tính tổng |ai-aj|, ta sort lại theo a tăng dần, dùng mảng tổng dồn s, khi đó tổng các |ai-aj| bị ảnh hưởng bởi i là s[n]-s[i]\*2-a[i]\*(n-i\*2).

Đpt : O(N\*log2(N)).

Sub4 : n<=500000.

Ta làm tưởng tự như sub2 nhưng thay vì phải xét 4 trường hợp thì ta tách công thức như sub3 rồi tính từng phần.

Cách tính từng phần : làm như sub3 nhưng do ci khác nhau lên ta cần nhân luôn ci khi tính các tổng này, công thức là (s[n]-s[i]\*2-a[i]\*(n-i\*2))\*c[i]. (c[i] ở đây là tương ứng với a[i] sau khi sort).

Đpt : O(N\*log2(N)).

\* Có thể giảm độ phức tạp xuống O(N) bằng cách thay vì sort bằng hàm có sẵn, ta sử dụng đếm phân phối.

**Bài 2: Upgrade**

- Phương án tối ưu sẽ luôn là từ nút lá đến nút lá khác;

- DFS đánh số các nút lá, một cây con gồm các nút lá liên tiếp tương ứng một đoạn. Biểu diễn dữ liệu lên mặt phẳng: một dự án (i,j) tương ứng một hình chữ nhật; một đường đi từ nút lá u đến một nút lá v tương ứng một điểm (u,v). Việc tìm phương án tối ưu tương đương việc tìm một điểm, dùng sweep line.

**Bài 3: Coloring**

1) Có thể dùng luồng hai cận, coi mỗi hàng, mỗi cột là một đỉnh. Thông lượng cận trên và cận dưới vào của mỗi hàng được xác định bằng r/2 và r/2+r%2 (trong đó r là số ô trên hàng đó). Tương tự xác định thông lượng đi ra của mỗi cột. Từ hàng i đến cột j có thông lượng cận trên bằng 1 nếu có ô (i,j) cần tô. Tìm luồng tương thích.

2) Cũng có thể sử dụng thuật toán tham:

Bước 1: Tô 4 ô tương ứng là 4 góc của một hình chữ nhật, loại bỏ 4 ô này;

Bước 2: Tô 3 ô tương ứng là 3 góc của một hình chữ nhật, thêm ô giả là ô góc thứ 4 còn lại;

Bước 3: Tô nốt các ô trên hàng và cột lần lượt theo R/B, điều chỉnh lại các ô đã tô ở bước 2.

3) Cũng có thể đưa về đường đi Euler.

**Bài 6: Mirror**

- Để xác định được vùng nhìn thấy được qua gương đặt ở cạnh AB, lấy N’ đối xứng của N qua AB, xác định giao điểm của N’A, N’B với các cạnh của đa giác.

- Để xác định được nhanh các giao điểm có thể tìm kiếm nhị phân và tính diện tích dựa theo tổng tiền tố.