

Dàn ý buổi thảo luận: Vận dụng thiết kế thuật toán: Geometric Algorithms

Phần 1: Giới thiệu về các bài toán hình học và ứng dụng thực tế (5 phút)

- **Hỏi 1 câu mời mọi tay phát biểu (Nhanh 1-2 phút)**
- **1.1. Định nghĩa và tầm quan trọng:**
 - Thuật toán hình học là gì? Đơn giản là các bài toán liên quan đến hình học như điểm, đoạn thẳng, đa giác.
 - Vì sao quan trọng? Chúng đóng vai trò lớn trong khoa học máy tính và nhiều lĩnh vực khác.
- **1.2. Ứng dụng thực tế siêu thú vị:**
 - **Ngày xưa:** Rất nhiều nhà toán học đã nghiên cứu nhiều bài toán hình học.
 - **Đồ họa máy tính:** Phát hiện va chạm (như trong game!), dựng hình 3D, xử lý ảnh.
 - **Robot học:** Giúp robot tìm đường đi và tránh vật cản.
 - **Hệ thống thông tin địa lý (GIS):** Phân tích không gian, tìm đường đi ngắn nhất (giống như Google Maps vậy).
 - **Competitive Programming:** Các bài toán tối ưu, tìm kiếm trên các đối tượng hình học.
 - Và còn nhiều ứng dụng khác nữa!

Phần 2: Đặc điểm và cách nhận diện các bài toán hình học (10 phút)

- **Cho thảo luận tìm ra các đặc điểm của các bài toán hình học (5 phút)**
- **2.1. Các đối tượng hình học cơ bản và tính chất:**
 - **Đối tượng:** Điểm, đoạn thẳng, đường thẳng, đa giác, vòng tròn...
 - **Tính chất quan trọng:**
 - Vị trí tương đối (giao nhau, song song, nằm trong/ngoài).
 - Khoảng cách, diện tích, chu vi.
 - Tính lồi/lõm, bao lồi (convex hull).
- **2.2. Cách nhận diện bài toán hình học trong Competitive Programming:**
 - Thường có input/output liên quan đến tọa độ, điểm, đoạn thẳng, đa giác.
 - Yêu cầu tính toán khoảng cách, diện tích, giao điểm, bao lồi, vị trí tương đối.
 - Khi thấy bài toán yêu cầu làm việc với điểm, đường thẳng, đoạn thẳng, góc, đa giác, khoảng cách, diện tích và cần dùng các công cụ như CCW, tích trong, lượng giác, thì khả năng cao đó là bài hình học.
- **2.3. Các bài toán *transform-and-conquer* sang hình học:**
 - Phổ biến là tối ưu tuyến tính cho hàm quy hoạch động (Convex hull trick Optimization).
 - Các bài toán liên quan đến cặp số có thể nhìn dưới góc độ hình học Transform and conquer (ví dụ bài tập về nhà nhóm 3).

Phần 3: Vận dụng các phương pháp thiết kế thuật toán để giải 2 bài toán: Closest Pair và Convex Hull (40 phút)

- **3.1. Bài toán Closest Pair (Cặp điểm gần nhất) (20 phút)**
- **Cho các bạn thảo luận tìm ra cách giải bằng cách vận dụng các phương pháp đã học**
 - **3.1.1. Giới thiệu:** Tìm hai điểm gần nhau nhất trong một tập hợp các điểm.
 - **3.1.2. Phương pháp Brute Force (Vét cạn):**

- **Đặc điểm:** Thử hết mọi cặp điểm. Hàm mục tiêu và không gian nghiệm rõ ràng, độ phức tạp ổn với N nhỏ, cài đặt dễ.
 - **Ưu điểm:** Dễ hiểu, dễ code, luôn đúng.
 - **Nhược điểm:** Chậm lắm nếu N lớn ($O(N^2)$).
 - **3.1.3. Phương pháp Divide and Conquer (Chia để trị):**
 - **Đặc điểm:** Chia nhỏ bài toán, tìm lời giải cho từng phần rồi kết hợp lại. Có tính độc lập con, cách kết hợp các bài toán con hiệu quả, việc giải quyết cục bộ cho một bài toán con nhỏ đơn giản (sau khi chia N đủ nhỏ).
 - **Ưu điểm:** Nhanh hơn nhiều ($O(N \log N)$), hiệu quả cho N lớn.
 - **Nhược điểm:** Thiết kế và chứng minh hơi phức tạp hơn chút.
 - **3.1.4. Các phương pháp khác:** Thuật toán gần đúng Randomized (có thể giải bằng Heuristic trong thời gian trung bình $O(N)$), Sweepline
- **3.2. Bài toán Convex Hull (Bao lồi) (20 phút)**
- **Cho các bạn thảo luận tìm ra cách giải bằng cách vận dụng các phương pháp đã học**
 - **3.2.1. Giới thiệu:** Tìm đa giác lồi nhỏ nhất bao quanh tất cả các điểm.
 - **3.2.2. Phương pháp Brute Force (Vét cạn):**
 - **Đặc điểm:** Kiểm tra tất cả các cặp điểm có thể tạo cạnh bao lồi. Không gian nghiệm hữu hạn (chỉ gồm $nC2$ đoạn thẳng), dễ cài đặt.
 - **Ưu điểm:** Ý tưởng đơn giản.
 - **Nhược điểm:** Rất chậm ($O(N^3)$), chỉ dùng cho N siêu nhỏ.
 - **3.2.3. Các phương pháp khác (Graham Scan hoặc Monotone Chain):**
 - **Đặc điểm:** Sắp xếp điểm và dùng stack để xây dựng bao lồi.
 - **Ưu điểm:** Cũng hiệu quả ($O(N \log N)$), thường dễ code hơn Quickhull.
 - **Nhược điểm:** Yêu cầu sắp xếp điểm ban đầu.
 - **3.2.4. Phương pháp Divide and Conquer (Chia để trị) / Quickhull:**
 - **Đặc điểm:** Tìm các điểm biên, chia tập điểm và đệ quy tìm các cạnh.
 - **Ưu điểm:** Hiệu quả hơn ($O(N \log N)$).
 - **Nhược điểm:** Cài đặt có thể phức tạp.

Phần 4: Kết luận và Hỏi đáp (5 phút)

- **4.1. Tóm tắt kinh nghiệm khi gặp các bài toán hình học:**
 - Thuật toán hình học rất rộng và có nhiều ứng dụng.
 - Hiểu đặc điểm bài toán là chìa khóa để chọn phương pháp đúng.
 - Các phương pháp như Brute Force, Divide and Conquer đều có thể áp dụng, nhưng hiệu suất khác nhau rõ rệt.
- **4.2. Hỏi & Đáp.**