**Thi vấn đáp: Báo cáo khoảng 10 trang và Demo thực hành (60%)**

1. Chạy các chương trình và giải thích code và độ phức tạp thuật toán của các phương thức trong tất cả các lớp đã học
2. **Hoàn thành cài đặt 22 bài tập thực hành:** 
   1. Mô tả phần thực hiện và bổ sung
   2. Đoạn code bổ sung
   3. Chạy chương trình cắt giao diện đưa vào báo cáo

**Các phần cài đặt sau trình bày code đầy đủ, chạy chương trình cắt giao diện đưa vào báo cáo**

1. Cài đặt Binary Search lọc Transactions nằm trong danh sách hợp lệ.
2. Hoàn thiện cài đặt UF.
3. Hoàn thiện Queue, Stack: sử dụng danh sách móc nối. Hoàn thiện cài đặt Evaluate một biểu thức có ngoặc với các phép toán, bổ sung phép toán khai căn sqrt lấy từ lớp Math.
4. Cài đặt tính toán biểu thức hậu tố có bổ sung phép chia số thực. Thêm phép chia vào SimpleSuffix
5. Cài đặt Counting Inversions (trong Beyond) và chương trình ứng dụng chạy inversion cho mảng các String, ví dụ danh sách các tên sinh viên trong lớp (không được sắp xếp, nên tạo ra các inversion).
6. Cài đặt merge(Comparable[] a, Comparable[] aux, int lo, int mid, int hi) trong ClosestPairs (trong Beyond).
7. Cài đặt Vietnames Date, Sử dụng IndexMinPQ in ra danh sách sắp xếp các cạnh của một đồ thị có trọng số cạnh tăng dần. . Cài đặt IndexMaxPQ.
8. Đóng gói AtivitySelection và Job.
9. Tạo lớp Sinhvien: mỗi sinh viên có Tên (họ đệm để riêng), MaSV, Điểm TBC, ngày tháng năm sinh, quê quán.
   1. Lập danh sách lớp sắp xếp theo tên
   2. Lập danh sách lớp sắp xếp theo ngày sinh từ già đến trẻ
   3. Lập danh sách sinh viên theo Quê (các sinh viên cùng quê sắp xếp theo tên). Cho quê, đưa ra danh sách sinh viên cso quê đó.
   4. Đọc danh sách sinh viên toàn trường, lập danh sách TopM sinh viên có điểm trung bình chung cao nhất.
   5. Có bảng điểm của riêng sinh viên đó, cài đặt tính điểm trung bình chung
10. Hoàn thiện BST: Cài đặt phương thức get(Node x, Key key) và put(Node x, Key key, Value val) trong Binary Search Tree (BST trong Searching).

Sử dụng BST lưu giữ thông tin Sinh viên để tìm kiếm theo mã SV, theo số điện thoại.

1. Cài đặt băm Bảng Đăng ký học tín chỉ lưu giữ Sinh viên và Danh sách cặp học phần-tín chỉ đăng ký học HashTable<Sinh vien, ST <String(tenhocphan), Integer(Sotinchi)>>, trong đó hàm băm dựa trên băm Họ đệm tên và ngày sinh. Việc so sánh hai sinh viên bằng nhau cũng dựa trên Họ đệm tên và ngày sinh (không để ý đến thuộc tính mã Sinh viên).
2. Lập Bảng điểm sinh viên: Đọc các bảng điểm của các môn của lớp, Tổng hợp thành bảng điểm chung và tính điểm trung bình chung.
3. Một số bài tập nâng cao bổ sung theo yêu cầu của giáo viên: Giải thích các chương trình ứng dụng: TopM, SET, FrequencyCounter, BlackList, WhiteList, FileIndex, LookupIndex.
4. FileFrequencyIndex
5. Hoàn thiện Kruskal. Hoàn thiện Dijkstra: cài đặt hàm tạo trong Dijkstra (trong Graph1), chọn phần tử nhỏ nhất.
6. Cài đặt thuật toán Prim, Dijkstra đích.
7. Cài đặt trong Jobs lựa chọn công việc nào (Context).
8. Cài đặt Knapsack dưới dạng đóng gói (Implementor). Cài đặt lớp Knapsack như một lớp có thể tái sử dụng được, in ra tổng lợi nhuận và phương án tối ưu gồm các đồ vật chất vào ba lô.
9. Hoàn thiện Bellman-ford (trong Graph1), cài đặt tìm đường đi ngắn nhất từ mỗi đỉnh đến đỉnh đích, Bellman-Ford đích
10. Hoàn thiện bài toán 8 Queen đưa ra mọi lời giải và đóng gói Nqueens. Cài đặt phương thức solveNQUtil trong AllSolQueen trong Backtracking
11. Hoàn thiện bài toán Ford\_Fulkerson (trong Context), đưa ra luồng cực đại và lát cắt tối thiểu. Cài đặt Cập nhật đường tăng luồng trong hàm tạo của Ford\_Fullkerson trong Context. Thêm phương thức in ra dung lượng của Min cut và lát cut đó.
12. Các bài tập nâng cao:
    1. Dijkstra tìm mọi đường đi giữa hai điểm: DijkstraAllPairs
    2. Đóng gói Nqueens: SolutionN, Nqueens, Knapsack
    3. FileSearch: FileFrequencyIndex; File-Count: File, Count (so sánh); Query: chứa từ khóa words [], results <File-Count>.
    4. Edmond-Branching