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

A. Chạy các chương trình và giải thích code và độ phức tạp thuật toán của các phương thức trong tất cả các class đã học

1. **Hoàn thành cài đặt 23 bài tập thực hành:** 
   1. Mô tả thành phần dữ liệu và các phương thức của các class trong ứng dụng.

Viết tên, tính chất (public, static,...), ko viết cụ thể code

* 1. Chạy chương trình ứng dụng, cắt giao diện kết quả đưa vào báo cáo

1. Cài đặt IndexMaxPQ (folder sort) (xong?)

Đối tượng có chỉ số, giá trị; sắp xếp PQ theo giá trị; lấy ra chỉ số

1. Cài đặt Dijkstra đích (folder Graph1)

Cho đích, tìm đường đi ngắn nhất từ mỗi đỉnh đến đích (trọng số ko âm)

1. Cài đặt DijkstraAllPairs: tìm mọi đường đi giữa hai đỉnh. (folder Graph1) (xong)

Đồ thị có hướng, trọng số ko âm, tìm đg đi giữa 2 đỉnh bất kì

Có mảng dijkstra gốc

1. Cài đặt PrimMST (greedy algorithmII) (folder Graph1)
2. Cài đặt Bellman-Ford đích (Bellman-Ford\_T) (folder Graph1)

Cho đích, tìm đường đi ngắn nhất từ mỗi đỉnh đến đích (trọng số có thể âm)

Quản lý cạnh đi vào

EdgeWeightDigraph\_T: quản lý Bag của cái đi vào

1. Đóng gói ba lô có phương thức:

taobalo(): thuật toán của balo nằm trong đây, tonggiatri(), dovat(),

oke

1. Đóng gói Nqueens (folder BackTracking): SolutionN, Nqueens: Tạo đối tượngNQueens(n), InKetqua:1 kết quả là 1 hoán vị của các con hậu, ví dụ (5,3,7,2,4,8,6), in ra 96 hoán vị 8 phần tử; Insoluongketqua(),

1. Giải thích các chương trình ứng dụng: TopM, SET, FrequencyCounter, BlackList, WhiteList, FileIndex, LookupIndex.
2. Đóng gói: FileFrequencyIndex: các phương thức Đọc file, query(word)

Đưa ra từ, cho kết quả: file nào chứa từ đấy bn lần

Có addfile() : mỗi lần đọc 1 file thì cập nhật

Tạo đối tượng rỗng rồi dùng addfile() để thêm file txt

1. Cài đặt lớp Sinh vien Sinhvien: tính diem TBC, tinh diem TBC hoc ky, ky(kythu), Nhapdiemmon(Mon m)
2. DS Lop: Tao từ file Excel, Nhap diem mon: gọi đến Nhapdiemmon() ở trên , Tổng kết học kỳ (xét thi đua)

In 10 sinh vien có TBC của kỳ thứ ... cao nhất

1. TopM sinh vien: theo tieu chi Trẻ, Điểm TBC, Tên và họ đệm
2. DS sinh viên dùng Bảng băm HashMaps: HashMap<Student, MaSV>: gán mã sinh viên cho sinh viên khi nhập học

Sắp xếp theo họ đệm, tên, quê

Đọc file tạo đối tượng DSSinhVien

1. Giả sử cho danh sách sinh vien dang excel: sinhvien.csv, trong đó có các cột masv, Tên, họ đệm, Điểm TBC, ngày tháng năm sinh, quê quán (tỉnh). Đọc file tạo đối tượng DSLop và cài đặt thêm các phương thức
   1. Truy vấn: InDanhsachĐồnghương (tỉnh) đưa ra các sinh viên có quê là tỉnh đã cho.
   2. Truy vấn: IndanhsachSinhvien(diem) đưa ra các sinh viên có điểm TBC từ cao đến thấp, mà điểm TBC >= diem
2. Cài đặt Ford-Fulkerson-Cut

So với Ford-Fulkerson: Có thêm đối tượng “lát cắt” đi ra từ tập thứ nhất đến tập thứ hai

1. VietnamDate: bổ sung nhiều cách tạo đối tượng Date: ngày/tháng/ năm; ngày - tháng – năm, ngày … tháng … năm … oke
2. Sử dụng Inversion trong Project Beyond: Cho một file Text danh sách sinh viên, mỗi sinh viên 1 dòng. Danh sách sinh viên trên chưa được sắp xếp theo (tên, họ và đệm). Tìm số nghịch thể trong danh sách đã cho và sắp xếp lại theo thứ tự tên, họ đệm
3. Cài đặt tính toán biểu thức hậu tố có bổ sung phép chia số thực. Thêm phép chia vào SimpleSuffix trong Project BagQueueStack (xong)

Thêm Phép chia kế thừa Expression, Phép chia sử dụng Double

1. Dựa trên Closest Pairs trong Project Beyond cài đặt FarthestPair: tìm 2 diểm xa nhau nhất. (xong)
2. Cài đặt thuật toán Erliest-Finish-Time-First cho bài toán Interval (dựa trên file Job trong Context) Scheduling trong Bài giảng Greedy Algorithm 1.

Có công việc j: bắt đầu lúc sj kết thúc lúc fj

2 công việc *tương thích* nếu ko chồng chéo

Tìm tập con tối đa các công việc ko chồng chéo

Cách làm: sắp xếp công việc theo thời gian kết thúc, tập A; nếu tương thích, đưa vào A

1. Đóng gói xếp lịch Weighted Interval Scheduling: đọc trong bài giảng Dynamic programming 1 và dựa trên file Job trong Context .

Có công việc j: bắt đầu lúc sj kết thúc lúc fj , có trọng số vj

Tìm tập con chứa các công việc cho trọng số lớn nhất

1. FileSearch: FileFrequencyIndex; File-Count: File, Count (so sánh); Query: chứa từ khóa words [], results <File-Count>.

Sắp xếp các file theo số lần xuất hiện

File nào k có đủ số từ thì bỏ qua chỉ thấy file chứa full

Nch là tìm xem các file đó chứa cả 2 từ đó không

Nó sẽ tính theo tổng số lần xuất hiện của cả 2 từ Có file + số lần xuất hiện ta có thể dùng array để nó tự sort

khi muốn them từ , sẽ lôi các file ra và them từ vào cái bảng chứa số lần xuất hiện (ST or HashTable ) , cập nhật count (tổng số lần xh) , đẩy vào pair : mảng trung gian mới - file nào k có thì bỏ đi

1. Cài đặt Edmond-Branching