**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC** **CÔNG NGHỆ THÔNG TIN**

**ĐHQG – TPHCM**

**KHOA: KHOA HỌC MÁY TÍNH**

****

**MÔN HỌC**

BIỂU DIỄN TRI THỨC VÀ SUY LUẬN

**CS214.I11**

**BÁO CÁO ĐỒ ÁN**

**Giải toán Lý thuyết đồ thị**

* GV Hướng dẫn: ThS. Nguyễn Đình Hiển
* SV Thực hiện:

Huỳnh Đăng Khoa ------------- 14520422

Trương Ngọc Kha ------------- 14520401

Phạm Vinh Hiển ------------- 14520274

***TP.Hồ Chí Minh***

***Tháng 12 Năm 2017***

**Nội dung**

[***Lời nói đầu*** 3](#_Toc502276753)

[**Nhận xét của giảng viên** 4](#_Toc502276754)

[CHƯƠNG I: GIỚI THIỆU TỔNG QUÁT 5](#_Toc502276755)

[1.1. Mục tiêu đề tài 5](#_Toc502276756)

[1.2 Yêu cầu và chức năng 5](#_Toc502276757)

[1.3 Phạm vi đề tài 5](#_Toc502276758)

[1.4 Quy trình thực hiện 6](#_Toc502276759)

[CHƯƠNG II: CƠ SỞ LÝ THUYẾT 7](#_Toc502276760)

[2.1 Dijkstra 7](#_Toc502276761)

[2.2 Kruskal 8](#_Toc502276762)

[2.3 Prim 8](#_Toc502276763)

[CHƯƠNG III: XÂY DỰNG VÀ THIẾT KẾ 9](#_Toc502276764)

[\* Xây dựng thuật giải bằng mã giả 9](#_Toc502276765)

[CHƯƠNG IV: TỔ CHỨC CHƯƠNG TRÌNH VÀ CHẠY THỬ NGHIỆM 10](#_Toc502276766)

[4.1 Cấu trúc chương trình 10](#_Toc502276767)

[4.2 Môi trường hoạt động 10](#_Toc502276768)

[4.3 Giao diện chương trình 11](#_Toc502276769)

[4.4 Hướng dẫn sử dụng 12](#_Toc502276770)

[4.5 Thử nghiệm bài toán 14](#_Toc502276771)

[4.6 Phân công công việc của từng thành viên 16](#_Toc502276772)

[TÀI LIỆU THAM KHẢO 17](#_Toc502276773)

***Lời nói đầu***

*Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Trong suốt thời gian từ khi bắt đầu học tập ở giảng đường đại học đến nay chúng em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý Thầy Cô, gia đình và bạn bè.*

*Với lòng biết ơn sâu sắc nhất, em xin gửi đến quý Thầy Cô ở Khoa Khoa Học Máy Tính – Trường Đại Học Công Nghệ Thông Tin đã cùng với tri thức và tâm huyết của mình để truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học tập tại trường. Và đặc biệt, trong học kỳ này, Khoa Bộ Môn đã tổ chức cho chúng em được tiếp cận với môn học mà theo em là rất hữu ích đối với sinh viên ngành Khoa Học Máy Tính cũng như tất cả các sinh viên thuộc các chuyên ngành Khoa Học Kĩ Thuật khác. Đó là môn học* ***“Biểu diễn tri thức và suy luận”.***

*Em xin chân thành cảm ơn thầy* ***Đỗ Văn Nhơn*** *và thầy* ***Nguyễn Đình Hiển*** *đã tận tâm hướng dẫn chúng em qua từng buổi học trên lớp cũng như những buổi nói chuyện, thảo luận về đề tài. Nếu không có những lời hướng dẫn, dạy bảo của cô thì em nghĩ đề tài này chúng em rất khó có thể hoàn thiện được. Một lần nữa, chúng em xin chân thành cảm ơn cô.*

TP Hồ Chí Minh 12/2017

*(Nhóm sinh viên thực hiện)*

**Nhận xét của giảng viên**

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

# CHƯƠNG I: GIỚI THIỆU TỔNG QUÁT

## 1.1. Mục tiêu đề tài

Trong đề tài này, chúng em xây dựng chương trình giải toán Lý thuyết đồ thị với khả năng nhập đồ thị trực quan. Minh họa các thuật giải Dijkstra, Kruskal, Prim thông qua mã giả, hình ảnh đồ thị, kết quả từng bước của thuật giải.

Với giao diện thân thiện, dễ sử dụng, phù hợp cho các giáo viên, học sinh, sinh viên trong việc áp dụng Công nghệ Thông tin vào dạy và học.

## 1.2 Yêu cầu và chức năng

Yêu cầu chính của chương trình là đưa ra kết quả sau khi chạy các thuật toán Dijkstra, Kruskal, Prim. Thêm đó chương trình còn minh họa thuật giải qua từng bước bằng mã giả và hình ảnh.

Chức năng của chương trình sẽ được xoay quanh các vấn đề sau:

1. Đưa ra kết quả đồ thị (bằng hình ảnh) và chữ sau khi chạy thuật giải Dijkstra, Kruskal, Prim.
2. Thời gian giải tự động cần thực hiện trong khoảng thời gian ngắn.
3. Các thao tác nhập liệu cần nhanh gọn và dễ thực hiện.
4. Kết quả yêu cầu cần chính xác
5. Giao diện phần mềm trực quan, dễ sử dụng.
6. Có khả năng nâng cấp sau này để giải nhiều dạng toán hơn.

## 1.3 Phạm vi đề tài

Phạm vi của đề tài này được xác định là tri thức về giải toán Lý thuyết đồ thị .Các thuật giải Dijkstra, Kruskal, Prim .

Cụ thể:

a. Dijkstra:

* Input:   
  Xét đồ thị G =(X,E) với các cạnh có trọng số không âm. Ma trận trọng số L (với qui ước Lhk = +∞ nếu không có cạnh nối từ đỉnh h đến đỉnh k) và hai đỉnh x,y cho trước.
* Output:

Đường đi ngắn nhất từ x đến y.

b. Kruskal

* Input:

Một đồ thị có trọng số liên thông với tập hợp đỉnh V và tập hợp cạnh E (trọng số có thể âm). Đồng thời cũng dùng V và E để ký hiệu số đỉnh và số cạnh của đồ thị.

* Output:

*V*mới và *E*mới là tập hợp đỉnh và tập hợp cạnh của một cây bao trùm nhỏ nhất

c. Prim

Input và output tương tự như của thuật toán Kruskal

## 1.4 Quy trình thực hiện

Đề tài sẽ được thực hiện qua các bước:

# CHƯƠNG II: CƠ SỞ LÝ THUYẾT

## 2.1 Dijkstra

Cho 1 đồ thị có hướng G = (V, E) với các cạnh có trọng số không âm, có dữ liệu nhập vào là ma trận trọng số L và 2 đỉnh x, y cho trước. Việc ta cần làm là tìm đường đi ngắn nhất từ x đến y trong đồ thị G.

Việc chúng ta cần làm là chỉ ra đỉnh v bất kì sao cho x -> v là đường đi ngắn nhất. Ta gọi length[v] là giá trị đường đi ngắn nhất từ x -> v, có thể hiểu length[v] là giá trị đường đi ngắn nhất trong các đường đi từ đỉnh xqua các đỉnh trong tập hợp S (nếu có) rồi đến v.

Thuật toán:

Bước 1: Khởi tạo các mảng n phần tử: label, length, prev. Gán label[k] = 1, length[k] = -1 (inf), prev[k] = -1 với k chạy từ 0 -> n – 1. Gán length[first] = 0

Bước 2: Chọn đỉnh v trong mảng sao cho length[k] là nhỏ nhất. Sau đó gán label[k] = 0 (Đã đánh dấu)

Bước 3: Tạo vòng lặp với biến chạy k, xét nếu label[k] = 1 (Chưa đánh dấu) và có đường đi từ v -> k: Nếu length[k] > length[v] + trọng số từ v -> k hoặc length[k] = inf, có nghĩa là nếu ta tìm được 1 đường từ v -> k là nhỏ nhất, hoặc là chưa tìm được đường nào ngắn nhất (inf) => Gán length[k] = length[v] + trọng số v -> k, prev[k] = v (Tạo vết chân đỉnh trước đó).

Bước 4: Nếu label[last] = 0 (Đã đánh dấu đỉnh đến), kết thúc vòng lặp. Nếu không thì quay lại bước 2.

## 2.2 Kruskal

Cho đồ thị G=(X, E).

Bước 1: Sắp xếp các cạnh của đồ thị theo thứ tự trọng số tăng dần.

Bước 2: Khởi tạo T := Ø.

Bước 3: Lần lượt lấy từng cạnh thuộc danh sách đã sắp xếp. Nếu T+ {e} không chứa chu trình thì gán T :=T+{e}.

Bước 4: Nếu T đủ n-1 phần tử thì dừng, ngược lại làm tiếp bước 2.

## 2.3 Prim

Thuật toán xuất phát từ một cây chỉ chứa đúng một đỉnh và mở rộng từng bước một, mỗi bước thêm một cạnh mới vào cây, cho tới khi bao trùm được tất cả các đỉnh của đồ thị.

* Dữ liệu vào: Một đồ thị có trọng số liên thông với tập hợp đỉnh *V* và tập hợp cạnh *E* (trọng số có thể âm). Đồng thời cũng dùng *V* và *E* để ký hiệu số đỉnh và số cạnh của đồ thị.
* Khởi tạo: *V*mới = {*x*}, trong đó *x* là một đỉnh bất kì (đỉnh bắt đầu) trong *V*, *E*mới = {}
* Lặp lại cho tới khi *V*mới = *V*:
  + Chọn cạnh (*u*, *v*) có trọng số nhỏ nhất thỏa mãn *u* thuộc *V*mới và *v* không thuộc *V*mới (nếu có nhiều cạnh như vậy thì chọn một cạnh bất kì trong chúng)
  + Thêm *v* vào *V*mới, và thêm cạnh (*u*, *v*) vào *E*mới
* Dữ liệu ra: *V*mới và *E*mới là tập hợp đỉnh và tập hợp cạnh của một cây bao trùm nhỏ nhất

# CHƯƠNG III: XÂY DỰNG VÀ THIẾT KẾ

## \* Xây dựng thuật giải bằng mã giả

a. Dijkstra

|  |
| --- |
| DijkstraHeapA(G(V,E⃗ ),w,sG(V,E→),w,s):  for v←1v←1 to VV  d[v]←+∞d[v]←+∞  d[s]←0d[s]←0  for every neighbor vv of ss  d[v]←w(s→v)d[v]←w(s→v)  B←B← BuildHeap(V∖{s})(V∖{s}) ≪B≪B contains all vertices except s≫s≫  repeat  u←u← ExtractMin(B)(B) ≪d[u]=δ(s,u)≫≪d[u]=δ(s,u)≫  for every neighbor vv of uu  d[v]←min(d[v],d[u]+w(u→v))d[v]←min(d[v],d[u]+w(u→v))  DecreaseKey(B,v,d[v])(B,v,d[v])  until B=∅ |

b. Kruskal

|  |
| --- |
| Bước 1:  Sắp xếp các cạnh của đồ thị theo thứ tự trọng số tăng dần.  Bước 2:  Khởi tạo T:= Ø  Bước 3:  Lần lượt lấy từng cạnh thuộc danh sách đã sắp xếp. Nếu T+{e} không chứa chu trình thì gán T:=T+{e}.  Bước 4:  Nếu T đủ n-1 phần tử thì dừng, ngược lại làm tiếp bước 2. |

c. Prim

|  |
| --- |
| PrimHeap(G(V,E),wG(V,E),w):  pick any vertex uu  M[1,…,n]←{0,…,0}M[1,…,n]←{0,…,0}  d[1,…,n]←{∞,…,∞}d[1,…,n]←{∞,…,∞}  for each neighbor xx of uu  M[x]←uM[x]←u  d[x]←w(u,x)d[x]←w(u,x)  initialize heap HH with keys d[1,2,…,n]d[1,2,…,n]  T←∅T←∅  while T≠VT≠V ≪T≪T does not contain all vertices≫≫  u←u← ExtractMin(H)(H)  for each neighbor xx of uu  if x∉Tx∉T and w(u,x)w(u,x) < d[x]d[x]  M[x]←uM[x]←u  d[x]←w(u,x)d[x]←w(u,x)  DecreaseKey(H,x,d[x])(H,x,d[x])  T←T∪{(u,M[u])}T←T∪{(u,M[u])} ≪≪ add edge (u,M[u])(u,M[u]) to T≫T≫  return T |

# CHƯƠNG IV: TỔ CHỨC CHƯƠNG TRÌNH VÀ CHẠY THỬ NGHIỆM

## 4.1 Cấu trúc chương trình

Input

Xử lý Input

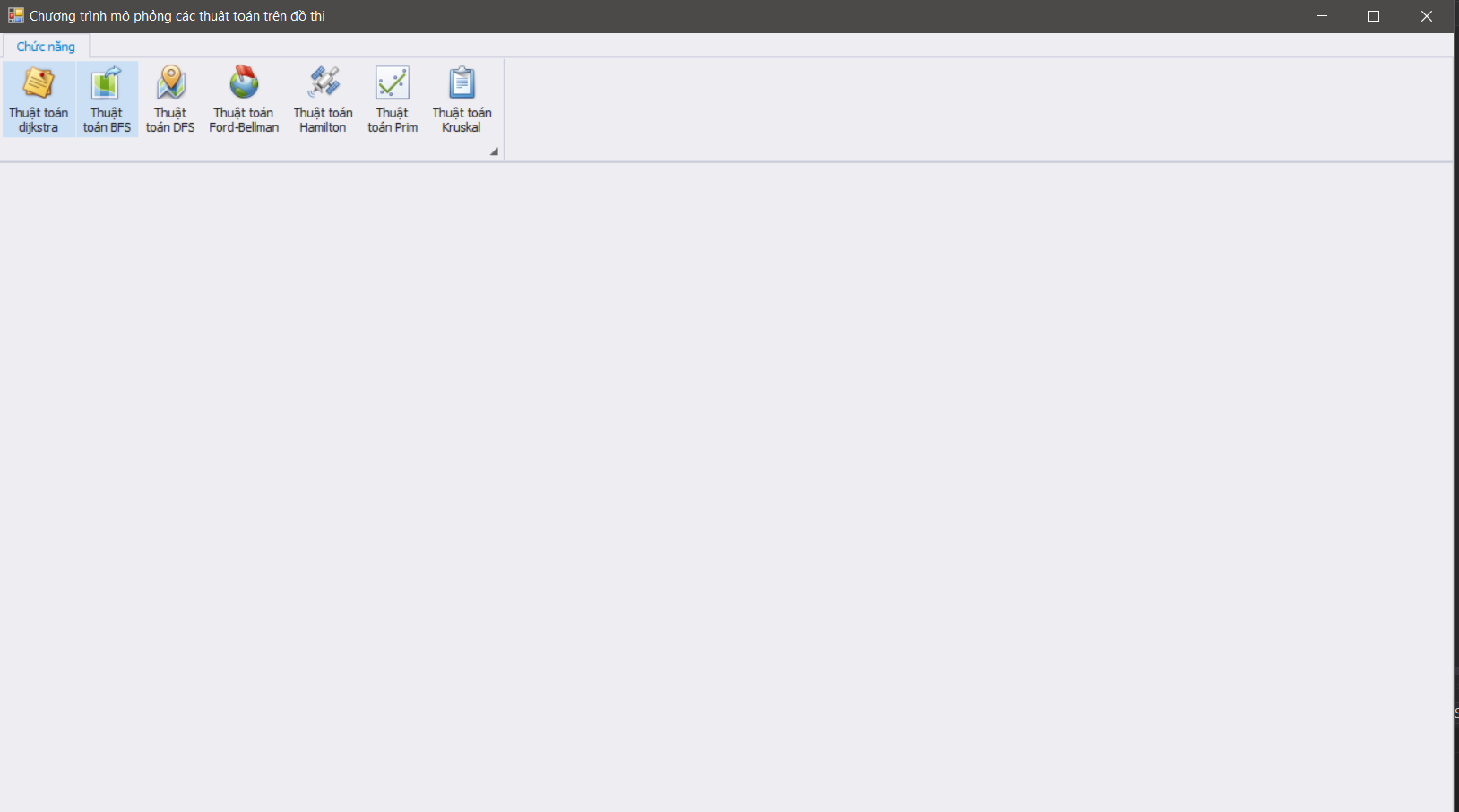
Giải quyết bài toán

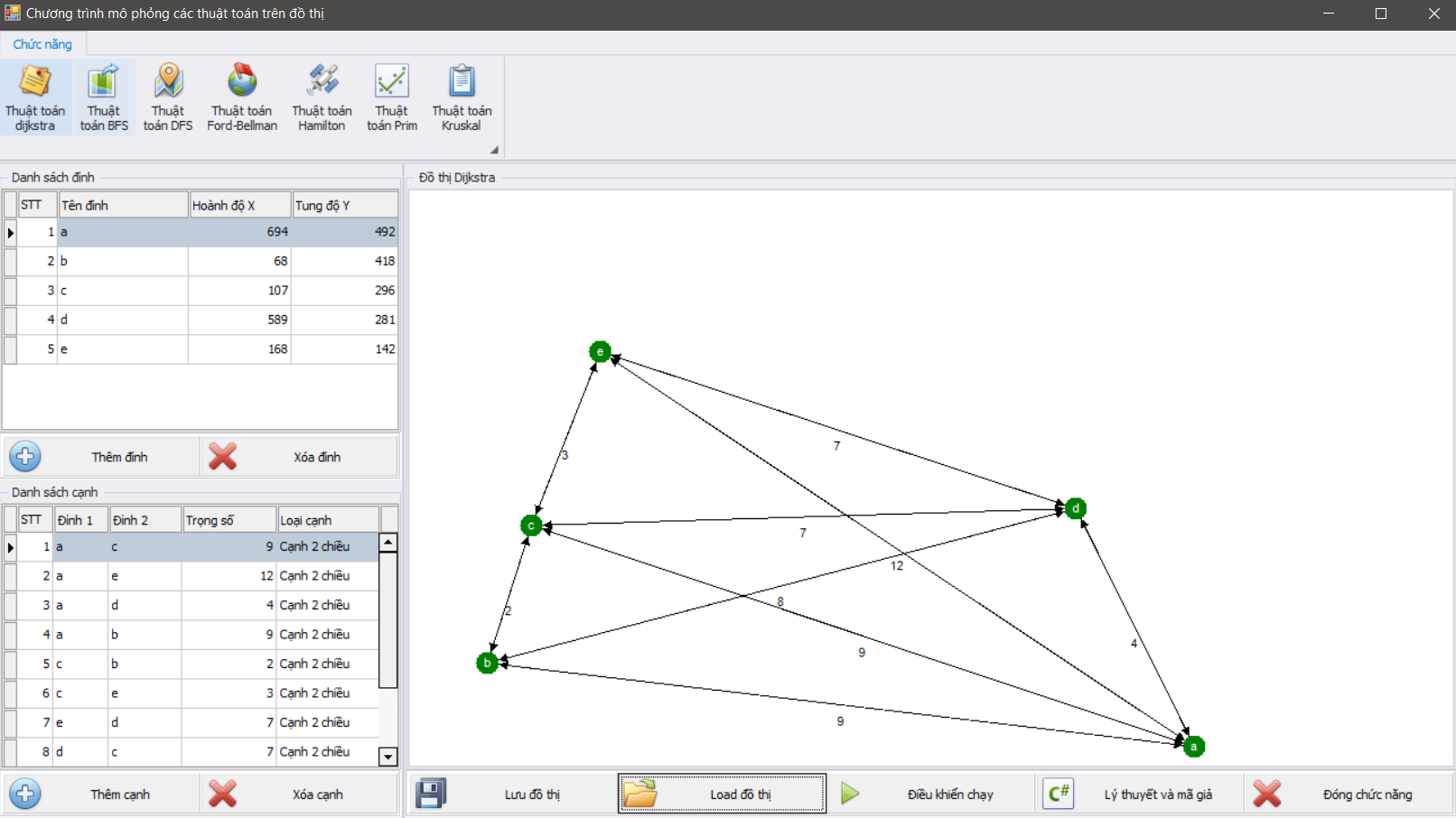
Output

## 4.2 Môi trường hoạt động

* Được viết bằng ngôn ngữ C# kết hợp với giao diện của DevExpress .
* Có thể chạy được trên hệ điều hành Windows 7 hoặc trở lên.

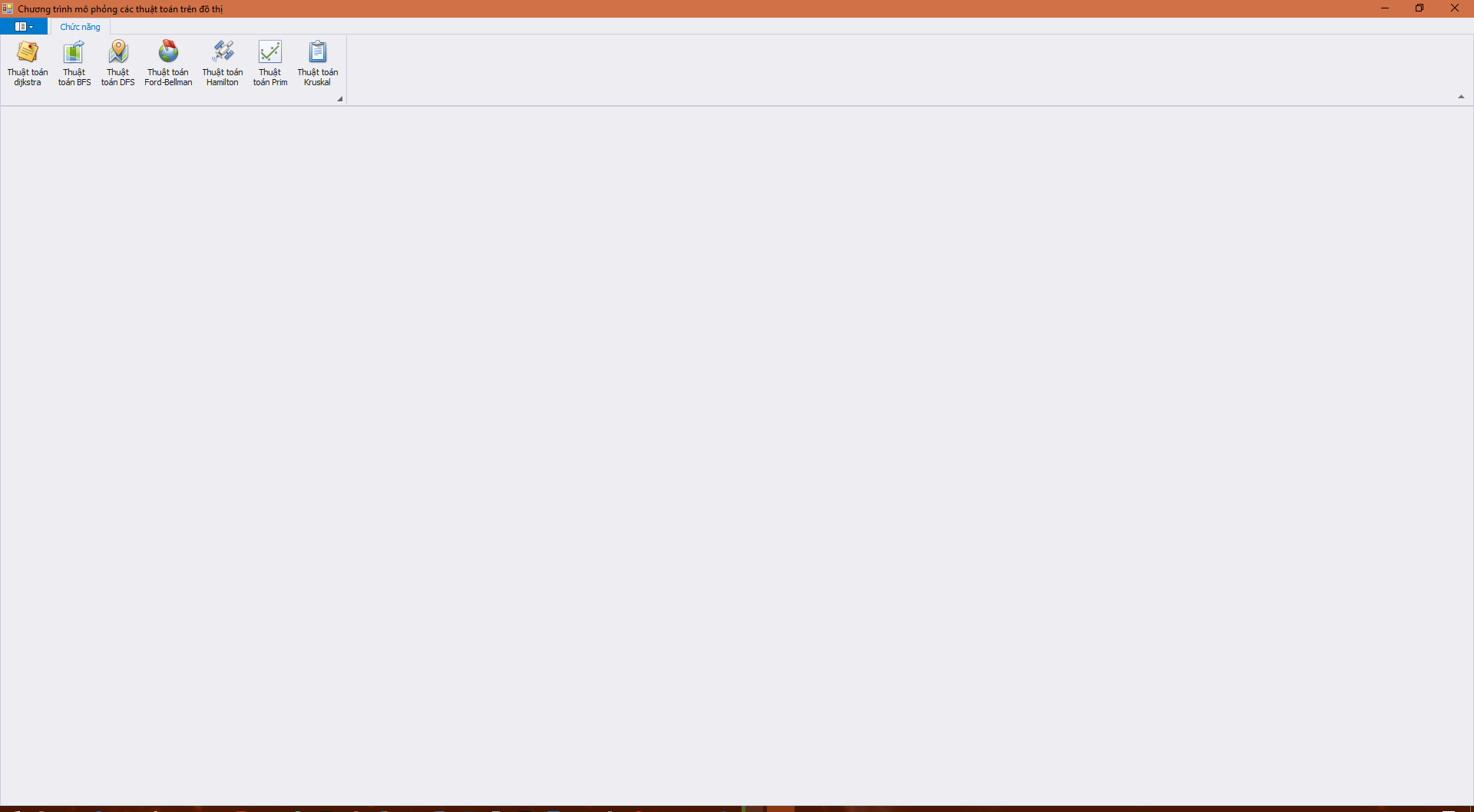
## 4.3 Giao diện chương trình

* Giao diện chính

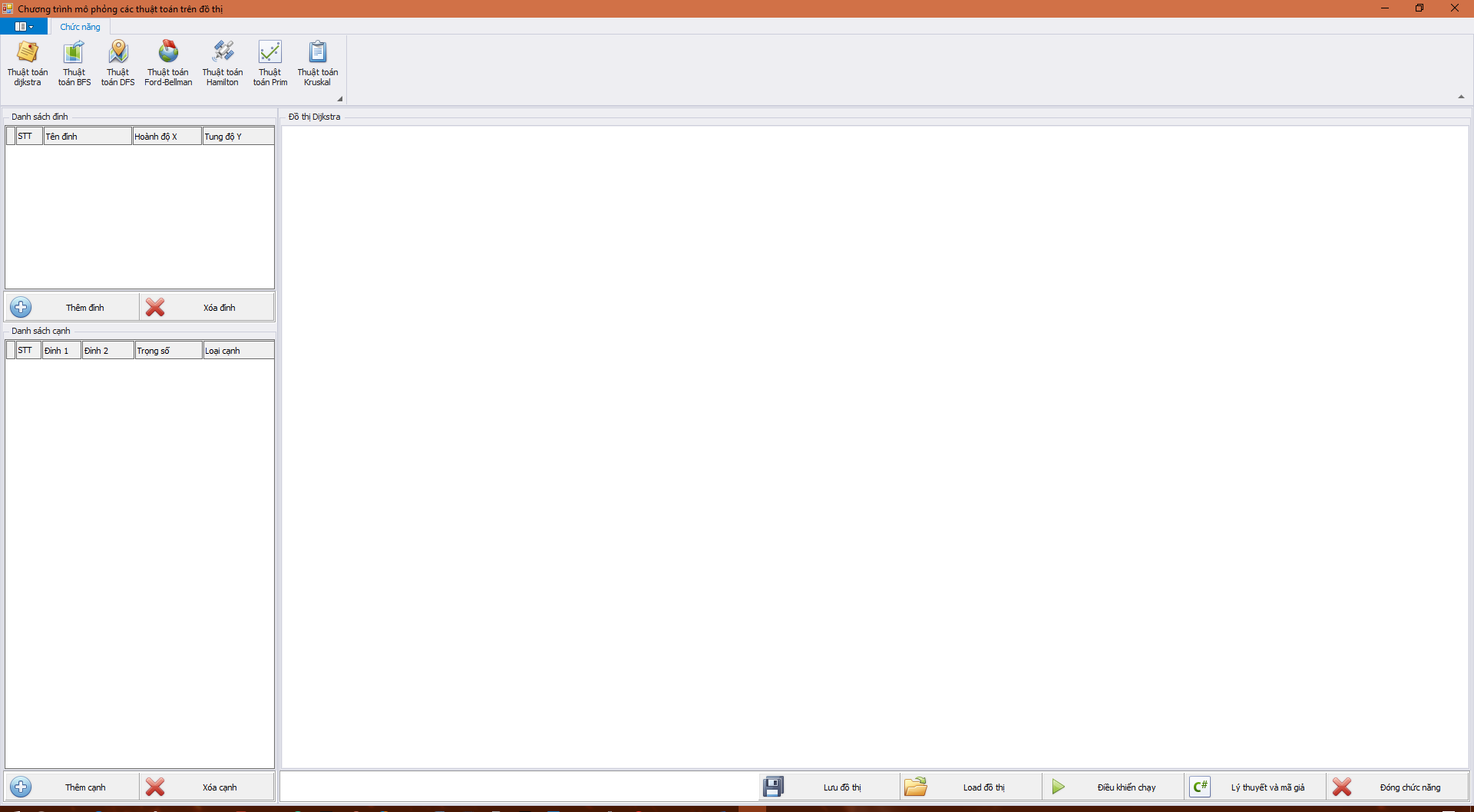


## 4.4 Hướng dẫn sử dụng

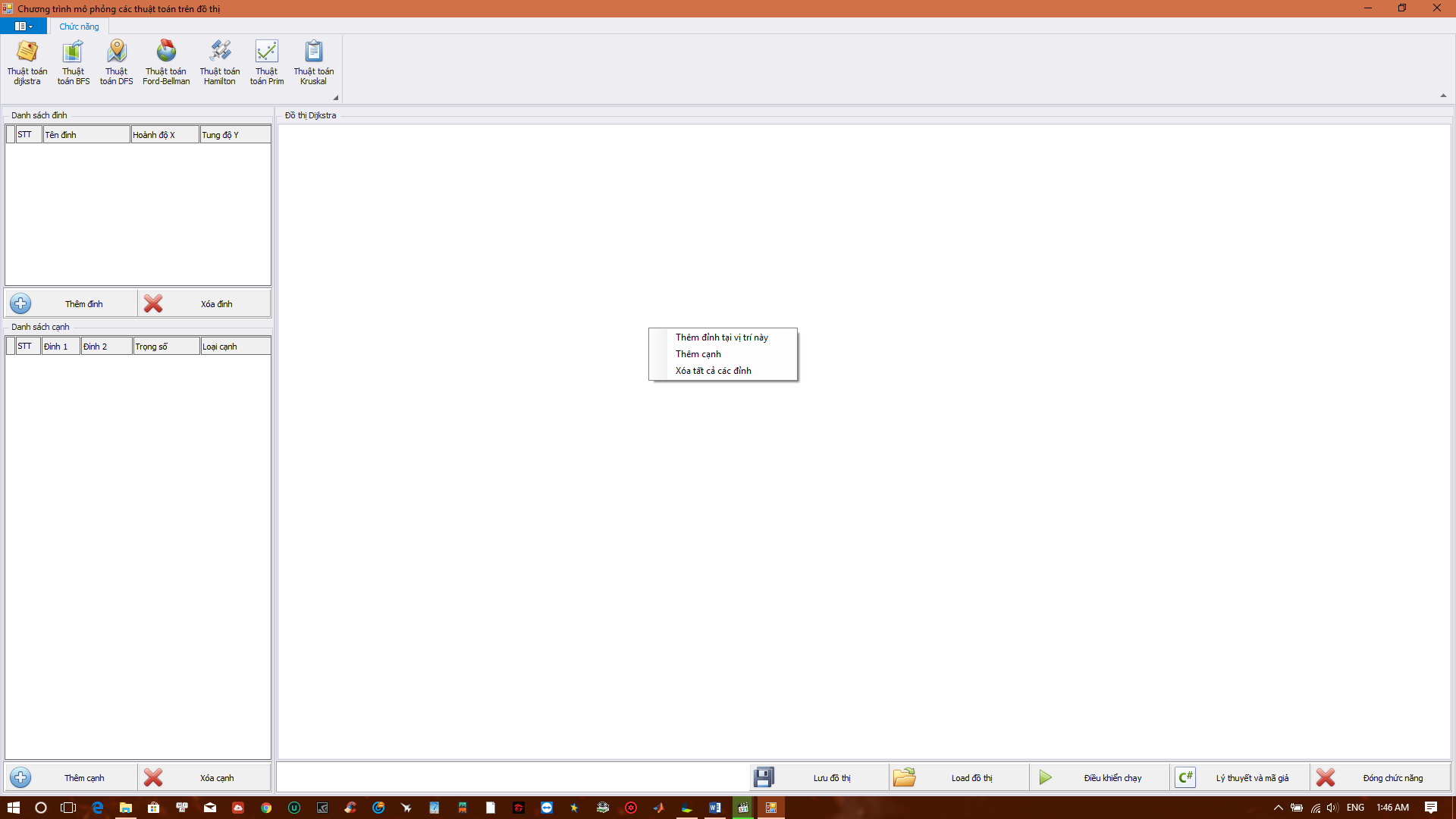
* Giao diện chính của chương trình



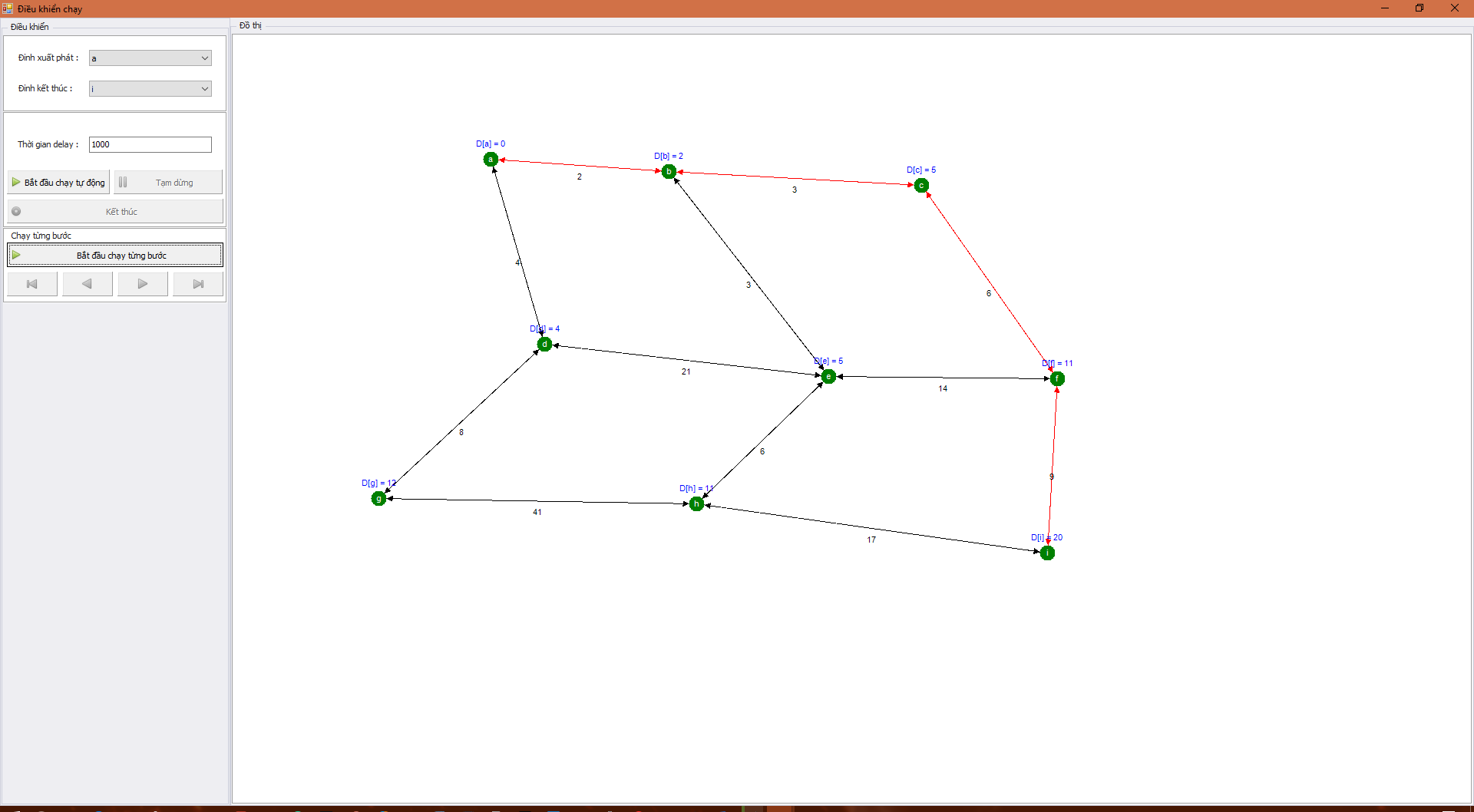
* Chọn thuật toán cần thực hiện , sau đó một bảng mói sẽ hiện ra



* 2 khung bên trái sẽ hiện thông tin các đỉnh và các cạnh
* Khung to bên phải dùng để vẽ đồ thị
* Nhấn chuột phải lên khung bên phải để hiện menu các chức năng vẽ đồ thị như Thêm đỉnh , Thêm cạnh , Xoá tất cả các đỉnh

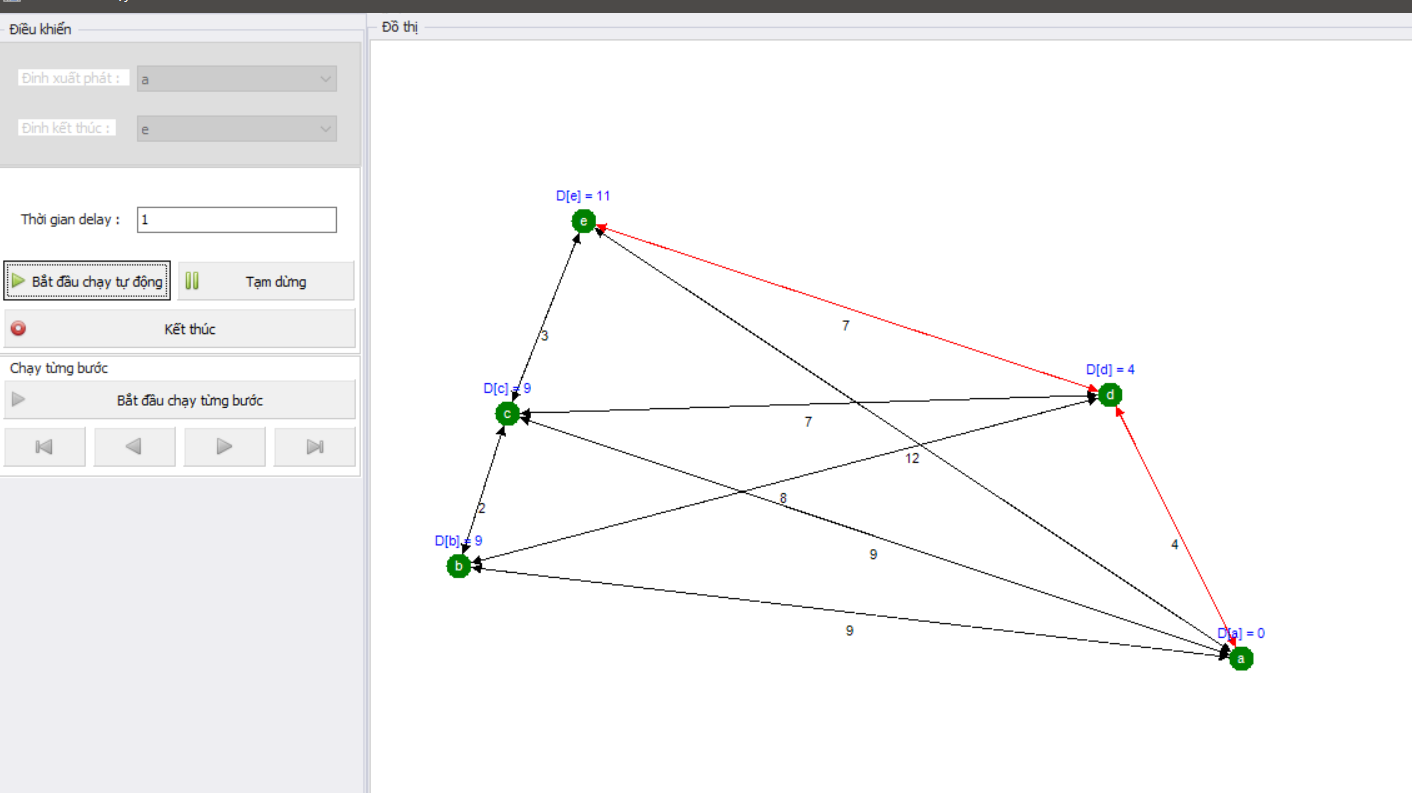


* Các phím chức năng bên dưới như
  + Lưu đồ thị : lưu đồ thị vừa vẽ được vào file txt
  + Load đồ thị : mở file txt đồ thị đã được vẽ sẵn
  + Điều khiển chạy : dùng để mở một form mới thực hiện quá trình chạy từng bước của thuật toán đã chọn
  + Lý thuyết và mã giả : hiện thông tin lý thuyết mã giả của thuật toán đã chọn
  + Đóng chức năng
* Màn hình điều khiển chạy :
* Chọn đỉnh bắt đầu và đỉnh kết thúc (tuỳ thuật toán)
* chạy liên tục với thời gian delay do người dùng nhập vào với n milisec hoặc chạy từng bước ở các nút chức năng bên dưới

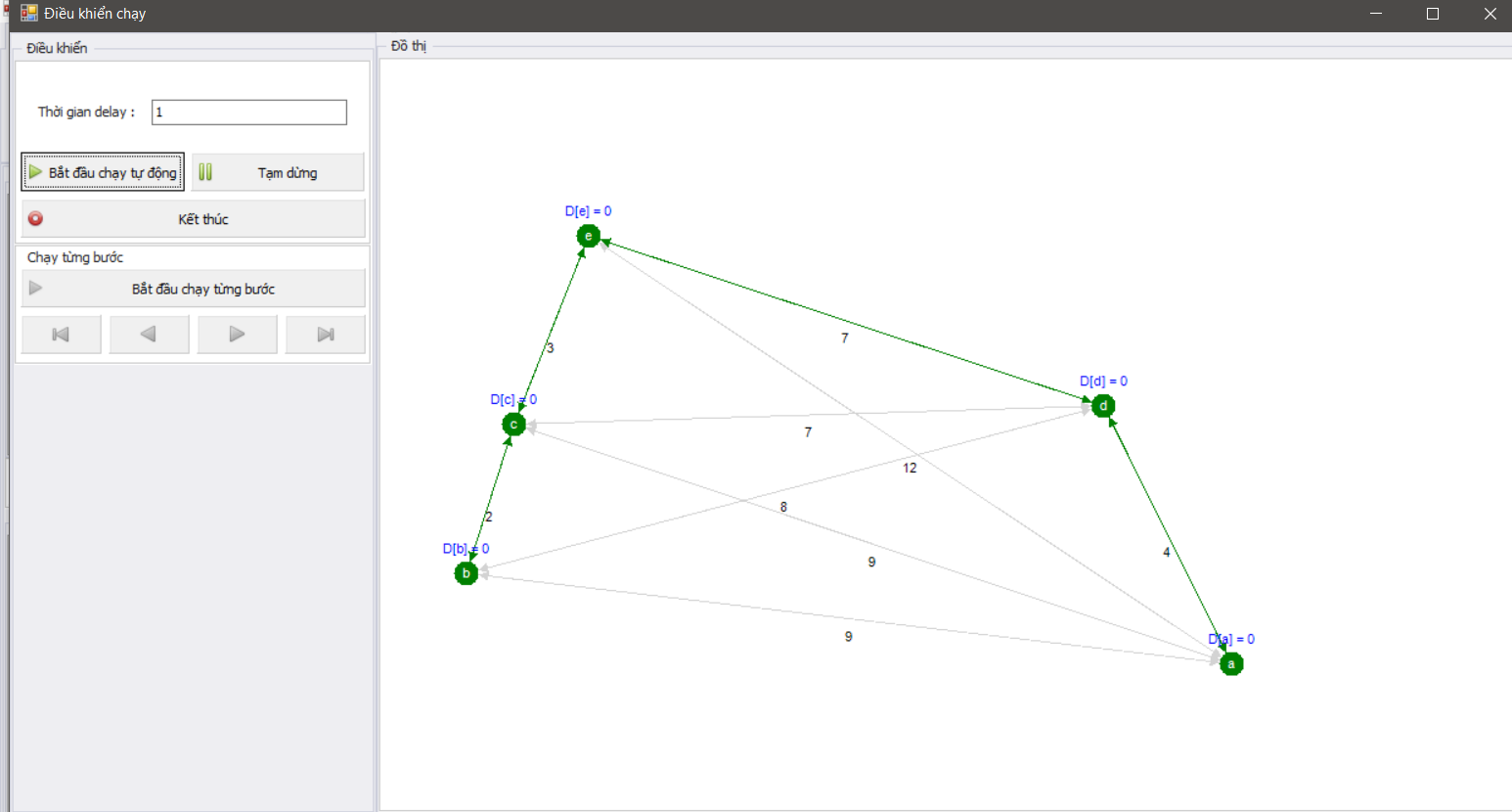


## 4.5 Thử nghiệm bài toán

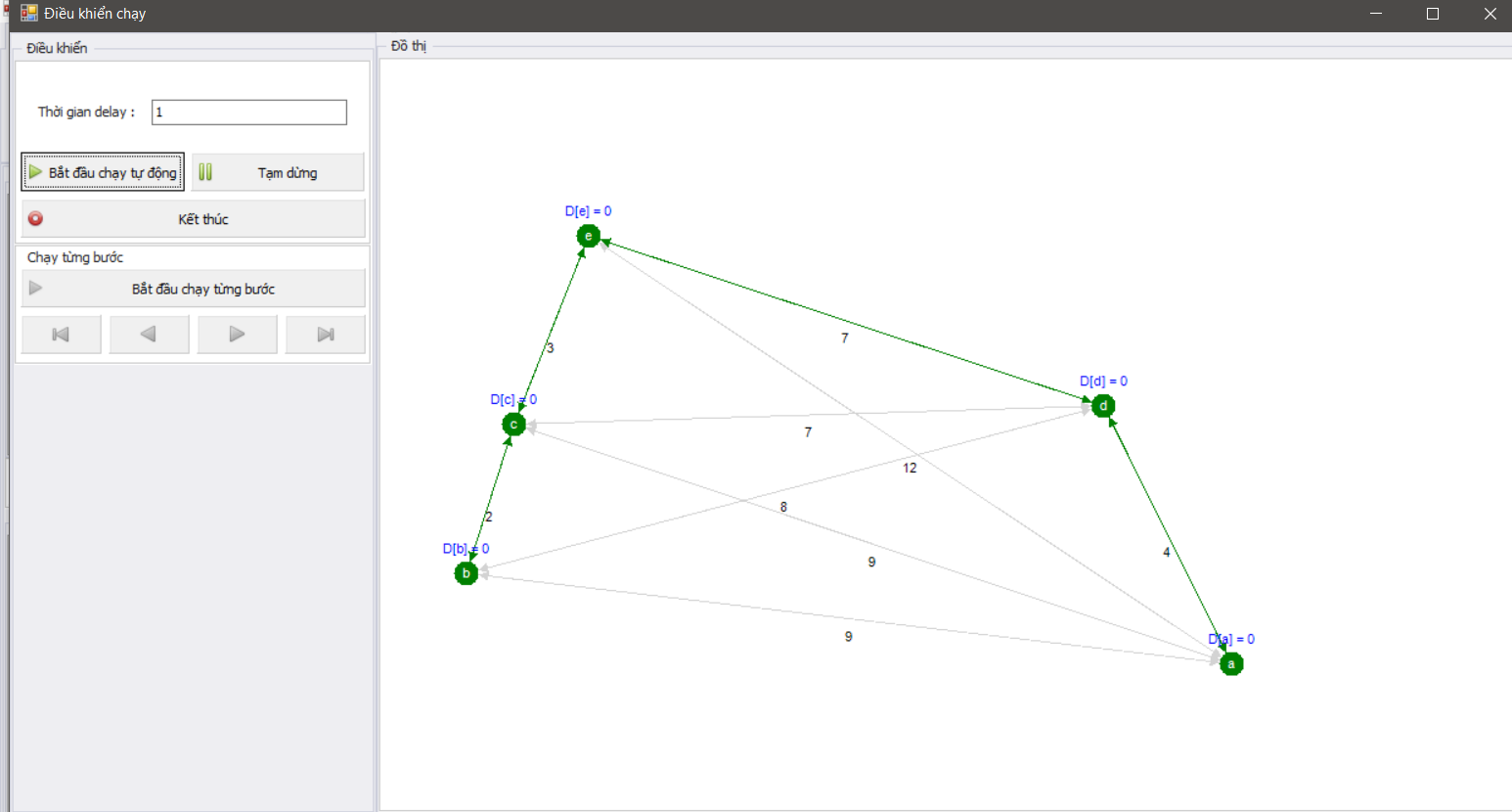
* ***Dijkstra***



* ***Kruskal***



* ***Prim***



## 4.6 Phân công công việc của từng thành viên

|  |  |  |
| --- | --- | --- |
| Họ và tên | MSSV | Công việc |
| Huỳnh Đăng Khoa | 14520422 | Thu thập tri thức, xây dựng giao diện, viết báo cáo. Sửa lổi kiểm thử. |
| Trương Ngọc Kha | 14520401 | Xây dựng thành phần xử lý. Xử lý input, output. Sửa lỗi kiểm thử. |
| Phạm Vinh Hiển | 14520274 | Xây dựng. Xử lý input, output. Sửa lỗi kiểm thử. |

# TÀI LIỆU THAM KHẢO

[1] Đỗ Văn Nhơn. **Giáo trình toán rời rạc.**

[2] Đỗ Văn Nhơn. **Biểu diễn tri thức và suy luận.**