Projecte Pràctic Part Greedy

Dijkstra

Estratègia greedy

Treball a realitzar

- Implementar dos versions del algorisme de Dijkstra
  - Algorisme de Dijkstra amb cerca seqüencial
  - Algorisme de Dijkstra amb cua o una altre estructura de dades que acceleri la cerca del vèrtex amb distancia mínima.
- Solució greedy al problema del viatjant de comerç
  - Modificar l'algorisme de Dijkstra perquè obtingui els camins mínims.
  - Implementar l'estratègia greedy d'anar sempre a la ciutat mes propera.
- Funcions on s'ha de implementar els algorismes
  - `void Dijkstra(CGraph& graph, CVertex *pStart)`
  - `void DijkstraQueue(CGraph& graph, CVertex *pStart)`
  - `CTrack SalesmanTrackGreedy(CGraph& graph, CVisits &visits)`
- Proves a superar:
  - Algorismes de Dijkstra: Entrada: grafs del directori TestDijkstra
  - Solució Greedy: Entrada: grafs i visites del directori TestSalesMan (revisar amb el corrector)

Algorisme de Dijkstra

- Inicialitzar les distancies dels vèrtex a infinit excepte la del vèrtex `pStart` que serà 0 (`double m_DijkstraDistance`).
- Marcar tots el vèrtex com no visitats (`bool m_DijkstraVisit`).
- `pActual` serà el vèrtex actual que l'inicialitzarem amb el vèrtex `pStart`
- Repetir
  - Recorre tots el veïns v de `pActual` i fer el següent
    - Si la distancia de v es més grossa que la distancia del vèrtex actual més la longitud de l'aresta que els uneix, actualitzar la distancia de v amb distancia de va més la longitud de la aresta que els uneix.
  - Marcar `pActual` com visitat
  - Si no hi ha més vèrtex no visitats finalitzar
  - `pActual`=vèrtex no visitat amb distancia més petita i tornar a repetir el bucle

Complexitat de l'algorisme de Dijkstra

- Graf amb V vèrtexs i A arestes
  - Algorisme seqüencial $O(V^2+A)=O(V^2)$
- Millora
  - Al cercar el nou vèrtex actual podem reduir els candidats a vèrtexs no visitats que son veïns de vèrtexs visitats.
  - Posem els vèrtexs candidats en una min-heap o una altre estructura que acceleri la cerca del que te la mínima distancia.
- Compte
  - El min-heap està ordenat per la distancia. Si fem un min-heap de vèrtexs no funciona, ja que l'algorisme de Dijkstra modifica les distancies i això corromp el min-heap.
  - Solució: fer un min-heap amb parelles vèrtex i la distancia calcula fins el moment.
  - No és solució treure del min-heap el vextex cada vegada que modifiquem la seva distancia, ja que aquesta operació no existeix per un min-heap.

Min-heap en C++

- La llibreria estàndard de C++ implementa un min-heap amb el template priority_queue
  - priority_queue<tipus elements, contenidor, comparador> queue
  - Tipus elements: tipus de dades dels elements que guardarà la cua.
  - Contenidor: On es guarda la cua: normalment un vector<tipus elements>
  - Comparador: classe utilitzada per compara els elements

```
struct comparador {
    bool operator() (element, element);
}
```

Complexitat Dijkstra amb cua

- Inicialitzar les distancies dels vèrtex a infinit excepte la del vèrtex start que serà 0 (`double m_DijkstraDistance`).
- Marcar tots el vèrtex com no visitats (`bool m_DijkstraVisit`).
- Posar a la cua el “vèrtex” start
- Mentre cua no buida
  - va=primer element de la cua i treure'l de la cua.
  - Si no està marcat com visitat
    - Recorre tots el veïns v de va i fer el següent
      - Si la distancia de v es més grossa que la distancia del vèrtex actual més la longitud de l’aresta que els uneix, actualitzar la distancia de v amb distancia de va més la longitud de la aresta que els uneix i posar-lo a la cua
    - Marcar va com visitat
- Complexitat: O((V+A)log(V+A))

Solució Greedy al problema del viatjant de comerç

- Sigui v la primera visita de la llista de visites
- Candidats= tots els vèrtex de la llista de visites menys el primer i l'últim.
- Repetir mentre candidats no estigui buit
  - Aplicant el algorisme de Dijkstra calcular les distancies de tots els vèrtexs del graf a v.
  - Seleccionar v1 com el vèrtex pertanyent a Candidats amb distancia mínima.
  - Afegir el camí de v a v1 al resultat.
  - Treure v1 de Candidats
  - v=v1
- Anar del últim candidat seleccionat al vèrtex final de la llista de visites.
- Com fer que el algorisme de Dijkstra ens doni els camins mínims:
  - Considerar que cada vegada que actualitzem una distancia d'un vèrtex, l'aresta per la que hem arribat el precedeix en el camí minin. Podem encadenar els camins afegint un apuntador a aresta a cada vèrtexs. Aquet l'actualitzem quan actualitzem la distancia del vèrtex.