データ構造とアルゴリズム演習　第12回クイズ

学修番号： ２２１４０００３

氏名： 佐倉仙汰郎

以下の問いに答えよ．

1. 「全域木」および「最小全域木」の定義を述べよ．説明のわかりやすさのために，表記や簡単なグラフの図を導入してもよい．

与えられた連結な無向グラフにおいて、すべての頂点を含みつつ、辺の重みの総和が最小となる木のこと

1. Dijkstra のアルゴリズムを，疑似コードを示しながら簡潔に説明せよ．説明のわかりやすさのために，表記や簡単なグラフの図を導入してもよい．

グラフ　G 　＝　（V,E）

Length(u,v)

／／初期化

各頂点に対し、ｄ（ｖ）←（ｖ＝ｓならば０，それ以外∞）

各頂点に対し、prev(v) ←なし

While(Q != NIL)

u ←Ｑからd(u)が最小である頂点を取り除く

for each ( u からの辺がある各頂点　ｖ)

if ( d(v) > d(u) + length(u,v))

d(v) ←d(u) + length(u,v)

prev(v) = u

1. 辺に重みのついた友向グラフのデータが手元にあるとする．そのグラフ上のあるノード を始点とする最短経路問題を解きたい．この問題に対する一般的な解法の候補は，「Bellman–Fordのアルゴリズム」か「Dijkstra のアルゴリズム」であり，手元の重み付き有向グラフの種類に応じて，これらのアルゴリズムを使い分けることが重要である．それぞれのアルゴリズムの特徴を，辺の重みの制約と計算量の観点から簡潔に説明せよ．

辺の重みの制約:

Bellman–Fordアルゴリズムは負の重みを持つ辺を含むグラフにも適用できる。

ただし、負の閉路が存在する場合は検出でき、最短経路が存在しないと判断できる。

計算量:

最悪の場合、計算量はO(V \* E)である。

ここで、Vは頂点の数、Eは辺の数。

特徴:

負の重みを含むグラフに対応できるため、特に負の重みが存在する可能性がある場合に有用。

Dijkstraのアルゴリズム:

辺の重みの制約:

Dijkstraアルゴリズムは、非負の重みを持つ辺を前提としている。

負の重みが存在する場合、正確な結果が得られない可能性がある。

計算量:

優先度付きキューを使用した場合、計算量は通常O((V + E) \* log(V))である。

ただし、Fibonacciヒープを使用することでより高速なO(E + V \* log(V))になることもある。

特徴:

非負の重みを前提としているが、その制約を満たす場合には通常より高速である。