CS 466: Design and Analysis of Algorithms

Michael Noukhovitch

Spring 2016, University of Waterloo

Notes written from Mark Petrick's lectures.

Contents

1 Travelling Salesman Problem

1.1 Definition

Given a graph $G = (V, E)$ with weights on edges $w : E \to \mathbb{R}^{+ve \cup \{0\}}$, find a **TSP Tour**:

- a cycle C that visits evert vertex exactly once
- has min $\sum_{e \in C} w(e)$

This is NP-Complete

1.2 Metrick TSP

Create an approximation problem with TSP by defining distances in a space:

- symmetry: $d(u, v) = d(v, u)$
- triangle inequality: $d(u, v) \leq d(u, w) + d(w, v)$

Approximation Algorithm (1977)

- find a minimum spanning tree and walk the tour
- take shortcuts to avoid re-visiting the same vertices
- by triangle inequality, the length is $\leq 2 * l_{MST}$ and therefore total $l \leq 2 * l_{TSP}$ (2-approx)
- ends up with runtime of MST, O(m log n)

1.3 Further Approximation

- TSP General: no constant factor approx
- Metric space: 1.5-approx, proven limit is 1.0045
- Euclidean space: $(1 + \epsilon)$-approx in polytime

2 Binomial Heap

2.1 Data Structures

Priority Queue

- store n elements each w/ integer key
- insert, deleteMin, decreaseKey, build (heapify)

Heap

- binary tree of elements
- structure (min key at root for min-heap)

- shape (almost complete binary tree, store in array)
- insert (+ bubbleUp) $O(\log n)$
- deleteMin $O(\log n)$
- decreaseKey $O(\log n)$
- heapify $O(n)$
- merge $O(n)$

2.2 Binomial Tree

2.2.1 Definition

- binomial tree of order 0 is a single node
- binomial tree of order k has a root node and its children are the binomial trees of order $k-1\dots 0$

2.2.2 Claims

- B_k has 2^k nodes
  $2^{k-1}+2^{k-1}=2^k$
- the height of B_k is k
  $k-1+1=k$
- there are $\binom{k}{i}$ nodes at depth i in B_k
  $\binom{k-1}{i}+\binom{k-1}{k-i}=\binom{k}{i}$

2.3 Binomial Heap

2.3.1 Definition

collection of binomial trees at most one of each rank (each one uses one heap order)

Example 2.1. Make a binomial tree of size n = 13
Expressing n in binary tells us what trees to build. 13 = 1101 so build $k=3$, $k=2$, $k=0$