Advanced Practical Programming for Scientists

Generated by Doxygen 1.8.8

Tue Jul 25 2017 11:46:53

Contents

Chapter 1

Exercise 10: Parallel Steiner Tree Heuristic

Extend exercise 9 such that it can be run in parallel (shared memory, e.g. OpenMP for C/C++). Furthermore, extend your program such that it tries each of the $***$first 100 terminals$***$ as starting points for the Steiner tree heuristic and keeps the best result. Your program should be executable as follows:

- prog <file.gph>

It should print the objective value (weight) of the best found Steiner tree, for instance

- Obj: 664

$***$Please note that this is the last exercise of the lecture and that it will play a major role in the final grading$***$

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 4

Class Documentation

4.1 graph Struct Reference

```
#include <ex10.h>
```

Public Attributes

- long long int number_of_nodes
- long long int number_of_edges
- long long int count
- long long int sum
- long long int ∗ number_of_neighbours
- long long int ∗ index_of_first_neighbour
- long long int ∗ sorted_heads
- long long int ∗ sorted_weights
- long long int ∗ tail
- long long int ∗ head
- long long int ∗ edge_weight
- long long int ∗ tree_pred
- long long int ∗ predecessor
- long long int ∗ distance

4.1.1 Detailed Description

the graphs attributes

4.1.2 Member Data Documentation

4.1.2.1 long long int graph::count

number of terminals added to subtree

4.1.2.2 long long int∗ graph::distance

the distance of each vertex from source

4.1.2.3 long long int∗ graph::edge_weight

the weight corresponding to each edge

4.1.2.4 long long int∗ graph::head

the head corresponding to each edge

4.1.2.5 long long int∗ graph::index_of_first_neighbour

index of first neighbour for each vertex

4.1.2.6 long long int graph::number_of_edges

number of edges in the graph

4.1.2.7 long long int∗ graph::number_of_neighbours

number of neighbours for each vertex

4.1.2.8 long long int graph::number_of_nodes

number of vertices in the graph

4.1.2.9 long long int∗ graph::predecessor

the predecessor of each vertex in shortest path tree

4.1.2.10 long long int∗ graph::sorted_heads

heads of each edge sorded by vertex

4.1.2.11 long long int∗ graph::sorted_weights

weights of each edge sorded by vertex

4.1.2.12 long long int graph::sum

sum of weights in subtree

4.1.2.13 long long int∗ graph::tail

the tail corresponding to each edge

4.1.2.14 long long int∗ graph::tree_pred

the predecessor of each vertex in steiner tree

The documentation for this struct was generated from the following file:

- ex10.h

Chapter 5

File Documentation

5.1 ex10.c File Reference

A program for APPFS ex10.

```
#include <assert.h>
#include <ctype.h>
#include <limits.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/time.h>
#include <time.h>
#include <omp.h>
#include "ex10.h"
```

Include dependency graph for ex10.c:

Macros

- #define **EXT_SIZE** (1024∗1024)
- #define **MAX_LINE_LEN** 512
- #define **INTEGER** long long int
- #define **INTEGER_MAX** LLONG_MAX
- #define **error_exit**(msg) error_exit_fun(msg, __FILE__, __LINE__)

Functions

- void error_exit_fun (const char ∗const msg, const char ∗const file, const INTEGER lineno)
- int sift_up (INTEGER ∗heap, INTEGER ∗distance, INTEGER ∗index, INTEGER current)
- int sift_down (INTEGER ∗heap, INTEGER ∗distance, INTEGER ∗index, INTEGER current, const INTEGER size)
- int steiner (struct graph ∗G, struct graph ∗H, INTEGER ∗is_prime, INTEGER source)
- INTEGER get_primes (INTEGER ∗is_prime, INTEGER max)
- int main (int argc, const char ∗const ∗const argv)

5.1.1 Detailed Description

A program for APPFS ex10.

Author

Tri-Peter Shrive

5.1.2 Function Documentation

5.1.2.1 void error_exit_fun (const char ∗const *msg*, const char ∗const *file*, const INTEGER *lineno*)

utility function that simplifies error handling

Parameters

msg	message to be displayed
file	file name
lineno	line number

5.1.2.2 INTEGER get_primes (INTEGER ∗ *is_prime*, INTEGER *max*)

sets entry at index of prime numbers to 1

Parameters

is_prime	allocate memory for this array of size max and set the memory to zero
max	size of array is_prime, the largest number to be assessed for primality

5.1.2.3 int main (int *argc*, const char ∗const ∗const *argv*)

reads data from file storing nodes and weights in graph structure. then calls dijkstra's algorithm and assesses longest shortes path

5.1.2.4 int sift_down (INTEGER ∗ *heap*, INTEGER ∗ *distance*, INTEGER ∗ *index*, INTEGER *current*, const INTEGER *size*)

sifts an entry down through binary heap

Parameters

heap	nodes in heap
distance	distance of nodes in heap
index	index of nodes in heap
current	current position of node in heap
size	size of heap

5.1.2.5 int sift_up (INTEGER ∗ *heap*, INTEGER ∗ *distance*, INTEGER ∗ *index*, INTEGER *current*)

sifts an entry up through binary heap

Parameters

heap	nodes in heap
distance	distance of nodes in heap
index	index of nodes in heap
current	current position of node in heap

5.1.2.6 int steiner (struct graph ∗ *G*, struct graph ∗ *H*, INTEGER ∗ *is_prime*, INTEGER *source*)

calculates steiner tree for given graph and source terminal using dijkstra's algorithm

Parameters

G	static graph attributes
H	variable graph attributes
is_prime	array where entries are 1 when index is prime
source	source node

5.2 ex10.h File Reference

Definitions for APPFS ex10.

This graph shows which files directly or indirectly include this file:

Classes

- struct graph

Functions

- void error_exit_fun (const char ∗const msg, const char ∗const file, const long long int lineno)
- int sift_up (long long int ∗heap, long long int ∗distance, long long int ∗index, long long int current)
- int sift_down (long long int ∗heap, long long int ∗distance, long long int ∗index, long long int current, const long long int size)
- int steiner (struct graph ∗G, struct graph ∗H, long long int ∗is_prime, long long int source)
- long long int get_primes (long long int ∗is_prime, long long int max)

5.2.1 Detailed Description

Definitions for APPFS ex10.

Author

Tri-Peter Shrive

5.2.2 Function Documentation

5.2.2.1 void error_exit_fun (const char ∗const *msg,* const char ∗const *file,* const long long int *lineno*)

Parameters

msg	message to be displayed
file	file name
lineno	line number

5.2.2.2 long long int get_primes (long long int ∗ *is_prime,* long long int *max*)

sets entry at index of prime numbers to 1

Parameters

is_prime	allocate memory for this array of size max and set the memory to zero
max	size of array is_prime, the largest number to be assessed for primality

5.2.2.3 int sift_down (long long int ∗ *heap*, long long int ∗ *distance*, long long int ∗ *index*, long long int *current*, const long long int *size*)

sifts an entry down through binary heap

Parameters

heap	nodes in heap
distance	distance of nodes in heap
index	index of nodes in heap
current	current position of node in heap
size	size of heap

5.2.2.4 int sift_up (long long int ∗ *heap*, long long int ∗ *distance*, long long int ∗ *index*, long long int *current*)

sifts an entry up through binary heap

Parameters

heap	nodes in heap
distance	distance of nodes in heap
index	index of nodes in heap
current	current position of node in heap

5.2.2.5 int steiner (struct graph ∗ *G*, struct graph ∗ *H*, long long int ∗ *is_prime*, long long int *source*)

calculates steiner tree for given graph and source terminal using dijkstra's algorithm

Parameters

G	static graph attributes
H	variable graph attributes
is_prime	array where entries are 1 when index is prime
source	source node