acm.mipt.ru

олимпиады по программированию на Физтехе

Раздел «Алгоритмы» . TopologicalSortCPP :

Пример кода топологической сортировки

Формулировка задачи и теория

Логика такая:

- dfs(A) = "напечатать все элементы одежды, которые должны идти до A и еще не напечатаны".
- getid(char[] term) - если слово term новое, то добавляет его в словарь terms; возвращает его индекс в массиве terms;

```
/*
    "Topological sort"

INPUT:
    Number of edges E followed by E lines.
    Each line has "edge description" — two words of length < 20, th

    EXAMPLE:
    IN:
4
sport medecine
medecine science
medecine health
knowledge science

    OUT:
sport
medecine
knowledge
science
health
*/


#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define M 15            // max term width
#define N 1000          // maximum number of terms
#define NE 100          // maximum edges from one vertex
#define P 10007         // hash table size
#define MAX_LINE_LENGTH  100    // max line length in input

typedef struct
{
  int to;
  int w;
} edge;

/* edge's weights  */
edge a[N][NE];

/* ne[i] number of edges from i-th term  */
int ne[N];

/*  number of terms */
int n = 0;

/*  vetex color aray */
```

Поиск
Поиск
Раздел «Алгоритмы»
Главная
Форум
Ссылки
El Judge
Инструменты:
Поиск
Изменения
Index
Статистика
Разделы
Информация
Алгоритмы
Язык Си
Язык Ruby
Язык Ассемблера
El Judge
Парадигмы
Образование
Сети
Objective C
Logon>>

```
int v[N];

/*  color for colouring component */
int c;

/*  hash table */
int hash_table[P];

/*  terms */
char term[N][M];

/*  number of found vetexes from which we have already exit */
int count;

/* returns id of the term (index of term in aray term) */
int
getid (char *s)
{
  unsigned long i, h1 = 0, h2 = 0;

  /* calculate two hash values */
  for (i = 0; s[i]; i++)
    {
      h1 *= 13;
      h1 += s[i] % 13;
      h2 *= 17;
      h2 += s[i] % 17;
    }
  h1 %= P;
  h2 %= P - 1;
  h2++;              // h2 should be >0 and  < P

  while (hash_table[h1] != -1)
    {
      /* collision or term is already known ?  */
      if (strcmp (term[hash_table[h1]], s) == 0) return hash_table[h1];
      // it is collision
      h1 += h2;
      h1 %= P;
    }

  /* we have new term - add then to array terms*/
  hash_table[h1] = n;
  strcpy (term[n], s);
  return n++;
}

/* DFS procedure. Outputs id in proper order */
void
dfs (int root)
{
  int i;
  v[root] = c;
  for (i = 0; i < ne[root]; i++)
    if (v[a[root][i].to] == 0)
      dfs (a[root][i].to);
  printf ("%s\n", term[root]);
}


int
main ()
{
  int i, j, id1, id2, m;

  /* Weight of the edge  */
  int w;

  /* Last scanned terms */
```

```
  char term1[M];
  char term2[M];
  char in[MAX_LINE_LENGTH];

  for (i = 0; i < P; i++) hash_table[i] = -1;
  for (i = 0; i < N; i++) ne[i] = 0;

  while (fgets (in, MAX_LINE_LENGTH, stdin) != NULL)
    {
        if(sscanf (in, "%s%s", term1, term2) == 2)
        {
          id1 = getid (term1);
          id2 = getid (term2);
          a[id2][ne[id2]].to = id1;
          a[id2][ne[id2]].w = w;
          ne[id2]++;
        }
    }

  c = 0;
  for (i = 0; i < n; i++) v[i] = 0;
  for (i = 0; i < n; i++) if (v[i] == 0)
     {
        c++;
        dfs (i);
    }

  return 0;
}
```

-- ArtemVoroztsov - 18 Mar 2004