Дедлайн: 23.03.2024 23:59

Инструкция

Задания в рамках домашней работы подразделяются на два блока:

1. Блок А «*Аналитические задачи*» – решения заданий А1, А2, ... оформляются в письменном в виде в любом удобном формате (TeX, скан и др.) и загружаются в соответствующие формы LMS.
2. Блок P «*Задачи на разработку*» – решения заданий P1-P5, ... загружаются в систему Яндекс.Контест и проходят автоматизированное тестирование.

Блок А				**Блок P**						**Итого (текущий максимум)**
A1	**A2**	**A3**	**A4***	**P1**	**P2**	**P3**	**P4**	**P5***	**P6**	71 (86*)
12	15	11	8	7	7	6	6	7	7	

Задачи, помеченные *, не являются обязательными для решения (относятся к *бонусным*). Подтверждение представленных решений бонусных заданий обязательно сопровождается устной защитой.

Плагиат влечет за собой обнуление результатов у всех вовлеченных лиц.

Удачи!

Блок А. Анализ различных графовых алгоритмов

<u>Задание А1</u> (12 баллов) Три способа построения (вроде бы) минимального остова

Ниже приведены три алгоритма **ALG_1**, **ALG_2** и **ALG_3**, которые из заданного связного неориентированного графа $G = (V, E)$ выбирают некоторое множество его ребер T.

```
ALG_1(G)
отсортировать ребра графа G в порядке
невозрастания весов
T = E
foreach (e ∈ E в порядке невозрастания весов)
    if (T − {e} является связным графом)
        T = T − {e}
return T
```

```
ALG_2(G)
T = ∅
foreach (e ∈ E, выбранное случайным образом)
    if T ∪ {e} не имеет циклов
        T = T ∪ {e}
return T
```

```
ALG_3(G)
T = ∅
foreach (e ∈ E, выбранное случайным образом)
    T = T ∪ {e}
    if (в T имеется цикл из ребер c ⊆ T)
        e_max: ребро с максимальным весом в
               цикле c
        T = T − {e}
return T
```

1. (6 баллов) Для каждого из трех представленных алгоритмов опишите его наиболее *эффективную* по общей временной сложности реализацию, в особенности, с точки зрения используемых *структур данных* и операций на них. Обоснуйте оценки сложности.
   Представьте исходный код на языке C++ для каждой из соответствующих реализаций, в котором используемые структуры данных достаточно отразить на уровне интерфейса без деталей реализации.
2. (6 баллов) Для каждого из трех представленных алгоритмов определите, формируется ли в множестве ребер T *минимальное остовное дерево* исходного графа G. Обоснуйте свой ответ и приведите (контр)примеры.

Дедлайн: 23.03.2024 23:59

<u>Задание A2</u> (15 баллов) Кратчайший блиц

Ниже приведены вопросы по практическому анализу различных алгоритмов поиска кратчайших путей в графах. Ответы на данные вопросы должны быть обоснованы и дополнены подтверждающими или опровергающими примерами.

1. (3 балла) Предположим, что «длина» пути рассчитывается не как общая сумма весов ребер, а как их *произведение*. Модифицируйте алгоритм Дейкстры для поиска кратчайших путей по этому предположению. Для каких графов модифицированный алгоритм **`DijkstraMULT`**`(G, start)` будет обеспечивать корректный поиск таких кратчайших путей? Почему?

2. (5 баллов) Разработайте алгоритм **`RestoreGraph`**`(dist[][])`, который по заданной матрице кратчайших путей `dist` между всеми парами вершин графа $G = (V, E)$ восстанавливает его представление.
   Например, на выходе этого алгоритма может быть получен *список смежности* графа G. Вы можете выбрать любое представление для восстанавливаемого графа, за исключением списка ребер. Есть ли случаи, в которых восстановление графа по матрице `dist` невозможно? Почему?

3. (3 балла) В ядре реализации *алгоритма Флойда-Уоршелла* (для поиска кратчайших путей между всеми парами вершин в графе), представленного ниже, допущена ошибка. Приведите пример графа, для которого кратчайшие пути будут определяться неверно, а также трассировку работы этого алгоритма.

```
for i = 1 to n
    for j = 1 to n
        for k = 1 to n
            dist[i][j] = min(dist[i][j], dist[i][k] + dist[k][j])
```

4. (4 балла) Возможно ли определить такой ориентированный взвешенный граф $G = (V, E)$, в котором некоторое ребро (v_i, v_j) лежит как на кратчайшем пути из вершины $a \in V$ в вершину $b \in V$, так и на кратчайшем пути из вершины b в вершину a? Охарактеризуйте данный граф и определите, есть ли ограничения применимости известных алгоритмов поиска кратчайших путей для графа G.

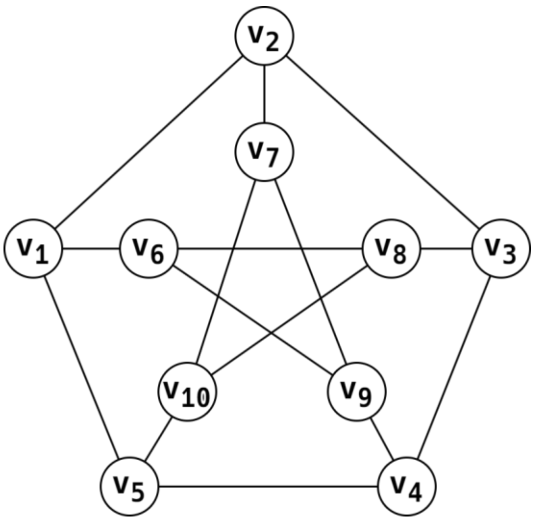
Задание A3 (11 баллов) Применение максимального потока

Задан ориентированный взвешенный граф $G = (V, E)$, в котором выделена вершина-исток $S \in V$, а также вершина-сток $T \in V$.

1. (8 баллов) Используя методы поиска максимального потока, разработайте алгоритм **`DISCONNECT(G,S,T)`** для поиска **минимального** количества вершин, которое необходимо удалить из исходного графа G, чтобы из вершины S стало невозможно достичь вершину T. Представьте обоснования, спецификацию, а также асимптотическую оценку верхней границы временной сложности алгоритма **`DISCONNECT(G,S,T)`**.
2. (3 балла) Представьте пример(-ы) работы алгоритма.

Задание A4* (8 баллов) Применение метода DELETION-CONTRACTION

Ниже приведен граф G, состоящий из 10 вершин и 15 ребер.



1. (7 баллов) Путем последовательного удаления ребер составьте **хроматический многочлен** $P(G, k)$ для заданного графа, где k – количество доступных цветов. В ответе представьте **все** промежуточные шаги.
2. (1 балл) Найдите **хроматическое число** $\chi(G)$ заданного графа.

Дедлайн: 23.03.2024 23:59

Блок P. Реализация графовых алгоритмов

Полная спецификация задач доступна в тестирующей системе Яндекс.Контест.

Задание P1 (7 баллов) Сортировка графа конденсации

В рамках этой задачи необходимо:

1. Найти компоненты сильной связности заданного ориентированного графа.
2. Выполнить топологическую сортировку графа конденсации.

Граф конденсации – это ориентированный ациклический граф, полученный путем «стягивания» каждой компоненты сильной связности исходного графа в одну вершину.

Задание P2 (7 баллов) Электроснабжение школ

В рамках этой задачи необходимо вычислить *две наиболее экономные* схемы электроснабжения школ при условии, что задана стоимость прокладки линии между парами школ.

Задание P3 (6 баллов) Алгоритм Беллмана-Форда

В данной задаче предлагается реализовать алгоритм Беллмана-Форда для поиска всех кратчайших путей из заданной вершины в остальные в ориентированном графе.

Задание P4 (6 баллов) Алгоритм Флойда-Уоршелла

В данной задаче предлагается реализовать алгоритм Флойда-Уоршелла для поиска кратчайших путей между всеми парами вершин в ориентированном графе.

Задание P5* (7 баллов) Числа-подстроки

За один шаг к числу X разрешается прибавить или из числа X разрешается вычесть любое положительное число Y, десятичная запись которого является подстрокой десятичной записи числа X. Стоимость такой операции равна сумме цифр числа Y.

Необходимо за минимальную стоимость получить из числа a число b, при этом все промежуточные числа должны быть положительными и не должны превышать n.

Задание P6 (7 баллов) Поиск максимального потока

В данной задаче требуется найти максимальный поток в графе, вершины которого пронумерованы от 1 до n.