**Graphs: Island Ferries**

The TopCoder website offers good advice on solving computer problems, including specifically graph problems. <http://www.topcoder.com/tc?module=Static&d1=tutorials&d2=alg_index>

(The Senior Computing Team participates in TopCoder Competitions.)

Here is one of TopCoder's graph problems. Write a solution. Good luck!

**IslandFerries**

|  |  |
| --- | --- |
| Problem Statement | |
|  | You are a traveler in a chain of tropical islands in the South Pacific. Transportation between these islands is provided by competing ferry services. One-way tickets for the ferries can be purchased on each of the islands. The tickets are only accepted by the ferry service for which the ticket was purchased. Each ticket entitles you to ride any single leg offered by that ferry service, where a leg is a trip between any two islands.  Ticket prices vary from island to island, and are published daily in the local newspaper. Armed with the current prices, you wish to decide which island to visit next. Your task is to determine for each island the cheapest path to journey from your current location to that island. You are not concerned with the number of ferries you must ride, only with the cost of the tickets required to make each journey.  Anti-competitive regulations prohibit you from carrying more than one ticket for the same ferry service, and from carrying more than three tickets total. Thus, after stepping off a ferry onto an island, you will have zero, one, or two tickets in your possession, and any tickets you are carrying will not be for the ferry you just rode. You begin with no tickets in your possession, so the first step of any journey must be to purchase one, two, or three tickets on the initial island (island 0).  The available legs will be given as a String[] **legs**, with each element giving the legs offered by a particular ferry service. The size of **legs** gives you the number of ferry services. Each element of **legs** will be formatted as a list of "<island1>-<island2>" (representing a one-way leg from <island1> to <island2>), each separated by a single space.  The prices of ferry tickets sold on each island is given as a String[] **prices**, with each element giving the prices for tickets on a particular island. The size of **prices** gives you the number of islands. Each element of **prices** will be a space-separated list of integers, corresponding to the ticket prices for ferries in the same order they are given in **legs**.  There will be up to 40 islands and 10 ferry services. Given the list of legs offered by each ferry service and the prices of tickets on each island, for each island compute the cost of traveling there from your initial island (island 0), and return the costs as a int[]. The size of your returned int[] should be one less than the number of islands. If a given island is unreachable, return -1 for the cost to that island. |
|  | |
| Definition | |
|  | |  |  | | --- | --- | | Class: | IslandFerries | | Method: | costs | | Parameters: | String[], String[] | | Returns: | int[] | | Method signature: | int[] costs(String[] legs, String[] prices) | | (be sure your method is public) | | |  | | |
| Notes | |
| - | There is not necessarily a route to all islands. |
| - | The cheapest route may including visiting an island more than once (see example 2 below). |
|  | |
| Constraints | |
| - | **legs** will contain between 1 and 10 elements, inclusive. The size of **legs** gives you F, the number of ferry services. |
| - | **prices** will contain between 2 and 40 elements, inclusive. The size of **prices** gives you I, the number of islands. |
| - | Each element of **legs** will contain between 1 and 50 characters, inclusive. |
| - | Each element of **prices** will contain between 1 and 50 characters, inclusive. |
| - | Each <island> in **legs** will be an integer between 0 and I-1, inclusive. |
| - | Each element of **prices** will contain F integers between 1 and 1000, inclusive. |
| - | There will be no duplicates within a given element of **legs**. |
| - | The *j*th element in the list of **prices[i]** gives the price of a ticket for ferry service *j* sold on island *i*. |
| - | No integers will have leading zeros. |
|  | |
| Examples | |
| 0) |  |
|  | |  |  |  | | --- | --- | --- | | |  | | --- | | { "0-1 0-3", "0-2" } | | { "5 7", "1000 1000", "1000 1000", "1000 1000" } | | | Returns: { 5, 7, 5 } | | |  | | --- | | You need an A ticket (cost of 5) to get to islands 1 and 3, and a B ticket (cost of 7) to get to island 2. | | |
| 1) |  |
|  | |  |  |  | | --- | --- | --- | | |  | | --- | | { "0-1 1-2 2-3", "0-1 2-3" } | | { "1 10", "20 25", "50 50", "1000 1000", "1000 1000" } | | | Returns: { 1, 11, 31, -1 } | | |  | | --- | | There are 5 islands and 2 ferry services. Referring to the ferry services as A and B, the cheapest way to get to island 1 is to purchase an A ticket on island 0 for a cost of 1 and take the A ferry to island 1.  The cheapest way to get to island 2 is to purchase A and B tickets on island 0 for a total cost of 11. Use the B ticket to get to island 1 and the A ticket to get to island 2.  The cheapest way to get to island 3 is to purchase A and B tickets on island 0 (cost of 11), use the A ticket to get to island 1, purchase another A ticket on island 1 (cost of 20), use it to get to island 2, and then use the B ticket to get to island 3. The total cost is 31.  There is no path to island 4. | | |
| 2) |  |
|  | |  |  |  | | --- | --- | --- | | |  | | --- | | { "0-1", "1-0", "0-2", "2-3" } | | { "1 1 1000 1000", "1000 1000 10 100", "1000 1000 1000 1000", "1000 1000 1000 1000" } | | | Returns: { 1, 12, 112 } | | |  | | --- | | The cheapest way to get to island 3 is to purchase A and B tickets on island 0, take the A ferry to island 1, purchase C and D tickets on island 1, take the B ferry *back* to island 0, then the C ferry to island 2, and then the D ferry to island 3. | | |
| 3) |  |
|  | |  |  |  | | --- | --- | --- | | |  | | --- | | { "1-0" } | | { "793", "350" } | | | Returns: { -1 } | | |  | | --- | | Legs are directed. | | |
| 4) |  |
|  | |  |  |  | | --- | --- | --- | | |  | | --- | | {"8-18 4-11 15-5 7-12 11-8 0-15 15-2 3-11 4-18 2-3",  "16-2 18-3 15-18 11-19 18-2 18-7 19-17 3-15 12-19",  "2-17 0-12 1-2 14-12 6-2 4-2 11-5 4-11 11-6 17-16",  "0-18 13-18 16-0 3-7 14-12 3-1 19-18 3-19 10-3 8-15",  "18-19 3-16 13-6 0-19 12-14 5-17 1-12 7-13 9-14 1-2",  "14-5 17-9 2-10 16-13 11-15 10-17 14-10 0-14 2-13",  "4-5 0-17 6-9 17-7 12-6 5-6 6-18 10-18 0-2 13-0 8-4",  "3-12 4-11 10-17 13-12 3-0 3-7 13-0 9-15 10-9 0-10" } | | {"592 219 88 529 324 86 503 610",  "2 94 8 600 34 95 6 494",  "638 301 10 246 290 97 85 74",  "118 8 939 393 450 79 317 99",  "99 806 698 740 2 26 525 818",  "95 9 615 972 23 23 5 666",  "6 448 440 710 83 4 419 496",  "4 47 182 4 185 70 718 770",  "3 321 6 855 968 65 10 6",  "173 224 300 3 79 2 707 49",  "21 10 7 10 4 9 5 8",  "6 600 4 724 7 1 960 247",  "83 16 901 50 437 780 658 2",  "763 923 125 576 45 423 3 1",  "7 324 391 898 8 59 281 973",  "9 44 49 364 78 744 43 5",  "10 62 75 418 216 90 32 919",  "764 534 778 605 80 647 821 74",  "65 449 102 867 421 94 6 7",  "67 155 362 789 189 316 107 595" } | | | Returns:  { 170, 160, 155, 173, 145, 150, 158, 168, 153, 145, 162, 88, 162, 86, 163, 159, 153, 150, 104 } | |  | | This problem statement is the exclusive and proprietary property of TopCoder, Inc. Any unauthorized use or reproduction of this information without the prior written consent of TopCoder, Inc. is strictly prohibited. (c) 2010, TopCoder, Inc. All rights reserved. | |

Hello and thanks for your inquiry.

You may use any TopCoder problem that's available in our archive for educational and non-commercial purposes.  There are a few requirements:

1) Any reproduction of a TopCoder problem statement must include any trademark or copyright statement from the original statement.  Proper credit must be given to TopCoder, including a link to the TopCoder website ([www.topcoder.com](http://www.topcoder.com/)).

2) The problem statements must be used "as is" with no modifications.

Regards,

Mike Lydon

CTO

TopCoder, Inc.

Here is a discussion of IslandFerries. Scroll down to the bottom of the page:

<http://www.topcoder.com/tc?module=Static&d1=match_editorials&d2=srm194>