

beCP 2022

Tâche 1.2: Rangemin (rangemin)

Auteur: Damien Galant

Preparation: Damien Galant, Victor Dheur, Pierre Vandenhove

Limite de temps: 1 s Limite mémoire: 512 MB

Note: Cette tâche est interactive. Veuillez consulter les instructions spéciales pour implémenter, compiler et tester votre programme. N'hésitez pas à demander de l'aide à un surveillant.

Il y a n entiers positifs distincts a_0, a_1, \dots, a_{n-1} . Votre but est de trouver l'indice du minimum parmi ces entiers. Cependant, vous n'avez pas directement accès à ces entiers, mais uniquement à la fonction `range_min(i)` (voir ci-dessous). Vous voulez appeler cette fonction *aussi peu de fois que possible*.

Fonctions à appeler

La fonction `range_min(i)` renvoie la *valeur* minimale parmi les entiers $a_i, a_{i+1}, \dots, a_{i+k-1}$. Une fois que vous avez trouvé l'indice de la valeur minimale parmi a_0, a_1, \dots, a_{n-1} , vous devez appeler la fonction `found_min(i)` une seule fois avec la valeur de cet indice puis arrêter votre programme.

Fonctions à implémenter

Vous devez implémenter la fonction `find_min(n, k)`. Cette fonction reçoit les valeurs de n et k . Vous devez appeler la fonction `range_min` pour trouver l'indice du minimum parmi a_0, a_1, \dots, a_{n-1} . Une fois qu'il est trouvé, vous devez appeler `found_min` une fois avec la valeur de cet indice puis arrêter votre programme.

Remarque : ne pas du tout appeler la fonction `range_min` pour une certaine entrée donnera un verdict de « mauvaise réponse » (« wrong answer »), même si vous trouvez le vrai indice du minimum (par pure chance).

Limites générales

Toutes les valeurs a_0, a_1, \dots, a_{n-1} sont des entiers *distincts* tels que $1 \leq a_i \leq 100\,000$ pour tout $0 \leq i < n$.

Vous pouvez appeler la fonction `range_min` au plus 100 000 fois (c'est une limitation pratique, une solution appelant `range_min` autant de fois recevrait de toutes façons un score très faible, voir la section "Scoring").

Contraintes supplémentaires

Sous-tâche	Points	Contraintes
A	20	$n = 100, k = 1$
B	20	$n = 100, k = 2$
C	60	$n = 1000, 1 \leq k \leq 500$

Exemple 1

Voici une suite possible d'évènements, en supposant que $n = 5, k = 2, a_0 = 8, a_1 = 10, a_2 = 4, a_3 = 5$ and $a_4 = 9$.

Votre appel	Réponse reçue
<code>range_min(0)</code>	$\min(8, 10) = 8$
<code>range_min(3)</code>	$\min(5, 9) = 5$
<code>range_min(2)</code>	$\min(4, 5) = 4$
<code>found_min(2)</code>	Indice trouvé!

Remarque : ce petit exemple n'est inclus dans aucune sous-tâches (parmi A, B et C).

Score

Le score est calculé de la façon suivante pour cette tâche :

- Pour obtenir des points pour une certaine sous-tâche, il est nécessaire que l'indice du minimum soit trouvé pour chaque test d'une sous-tâche.
- Lorsque c'est le cas, notons S le nombre d'appels à `range_min` réalisés par votre solution, et par B le nombre d'appels à `range_min` réalisés par la solution préparée par les juges de beCP. Le *ratio* pour ce test est défini comme $\min(B/S, 1)$. Sur CMS, le ratio sera affiché pour chaque test où l'indice du minimum a été correctement trouvé.
- Finalement, votre score pour une sous-tâche est donné par le nombre total de points pour cette sous-tâche, multiplié par $0.2 + 0.8 * r$, où r est le *plus petit* ratio parmi les tests de cette sous-tâche.

Par exemple, si le plus petit ratio parmi les tests de la sous-tâche B est égal à 0.4, alors votre score pour la sous-tâche B sera égal à $20 * (0.2 + 0.8 * 0.4) = 10.4$.

Information de compilation

Cette tâche est interactive. Suivez les étapes suivantes pour compiler et exécuter votre programme. Demandez de l'aide à un responsable si nécessaire.

- Ouvrez un terminal
- Pour afficher la liste des fichiers présents dans un dossier, utilisez la commande `ls`
- Pour changer de dossier courant (vers le dossier contenant les codes à modifier), utilisez la commande `cd nom.du.dossier`
- Pour compiler et exécuter des programmes écrits en C++, les commandes sont les suivantes :
 - Compilation :
`g++ -std=c++11 -Wall -Werror -Wshadow grader.cpp rangemin.cpp`
 - Execution : `./a.out < input.in`

Le résultat de votre programme sera affiché à la console : « FOUND THE INDEX AFTER X CALLS » si l'indice du minimum a été trouvé après X étapes, et « WRONG INDEX WAS GIVEN » si l'indice fourni est incorrect. D'autres messages peuvent être affichés si les arguments fournis à `range.min` ou `found_min` sont erronés. Si vous modifiez le fichier `input.in`, sachez que la première ligne contient n et k et que la seconde ligne contient les n entiers a_0, a_1, \dots, a_n . Assurez vous que ces entiers soient distincts et compris entre 1 et 100 000 !

Remarques

- Vous devez seulement soumettre le fichier `rangemin.cpp`. Il doit implémenter la fonction `find_min` décrite ci-dessus.
- **N'affichez rien à la console dans le programme que vous soumettez (`printf`, `cout`).** Vous pouvez utiliser `cerr` en C++ pour afficher de l'information de débogage.