Question

Slavic is preparing a present for a friend's birthday. He has an array a of n digits and the present will be the product of all these digits. Because Slavic is a good kid who wants to make the biggest product possible, he wants to add 1 to exactly one of his digits. \n\nWhat is the maximum product Slavic can make?\n\nInput\n\nThe first line contains a single integer $t(1 \leq t \leq 10^4)$— the number of test cases.\n\nThe first line of each test case contains a single integer $n(1 \leq n \leq 9)$— the number of digits.\n\nThe second line of each test case contains n space-separated integers a_i $(0 \leq a_i \leq 9)$— the digits in the array.\n\nOutput\n\nFor each test case, output a single integer— the maximum product Slavic can make, by adding 1 to exactly one of his digits.Sample Input 1:\n4\n\n4\n\n2 2 1 2\n\n3\n\n0 1 2\n\n5\n\n4 3 2 3 4\n\n9\n\n9 9 9 9 9 9 9 9 9\n\n\n\nSample Output 1:\n\n16\n2\n432\n430467210\n"

Ours Longest: 2932 Tokens

NRP: 2471 Tokens

Okay, I need to solve this problem where Slavic can add 1 to exactly one digit in an array of digits to maximize the product. Hmm, let's think about how to approach this.\n\nSo, the problem is: given an array of digits (each between 0 and 9), I can choose one digit to add 1 to. Then, compute the product of all the digits. I need to find the maximum possible product after this operation.\n\nFirst, I need to [omitted...]Sample Input 1:\n\n4\n\n4\n\n2 2 1 2\n\nSo, the array is [2,2,1,2]. For each position, adding 1 and compute product.\n\nLet's see:\n\nOption 1: add to first 2 → [3,2,1,2]. Product is 3*2*1*2=12.\n\nOption 2: add to second 2 → same as option 1, product 12.\n\nOption 3: add to 1 → [2,2,2,2]. Product is 2^4=16.\n\nOption4: add to last 2 → [2,2,1,3]. Product is 2*2*1*3=12.\n\nSo maximum is 16. Which matches the sample output.\n\nAnother sample:\n\n3\n\n0 1 2\n\nPossible options:\n\nAdd 1 to 0 → [1,1,2] → product 1*1*2=2.\n\nAdd 1 to 1 → [0,2,2] → 0*2*2=0.\n\nAdd 1 to 2 → [0,1,3] → 0*1*3=0.\n\nSo maximum is 2. Which matches sample.\n\nAnother sample: 5 digits 4,3,2,3,4.\n\nWhat's the best option?\n\nAdding 1 to the 2: makes it 3, so the array becomes [4,3,3,3,4]. Product is 4*3*3*3*4 = 4*3^3 *4 = 4*27 *4 = 432.\n\nWhich matches sample.\n\nAnother sample: 9 9s. Adding 1 to one of them gives 10, others 9. So product is 10 * 9^8.\n\nCompute 9^8: 9*9=81, 81*9=729, 729*9=6561, 6561*9=59049, 59049*9=531441, 531441*9=4782969, 4782969*9=43046721. Multiply by 10: 430467210. Which matches the sample.\n\nSo the code should handle all cases correctly.\n\nNow, in Python, the code can be written as follows.\n\nRead t:\n\n**t = int(input())\n\nfor _ in range(t):\n\n n = int(input())\n\n a = list(map(int, input().split()))\n\n max_prod = 0\n\n for i in range(n):\n\n new_a = a.copy()\n\n new_a[i] += 1\n\n product = 1\n\n for num in new_a:\n\n product *= num\n\n if product > max_prod:\n\n max_prod = product\n\n print(max_prod)**\n\nYes, that's the code.

Remaining: None

NRP Ratio: 100%

AdaptThink Longest: 3797 Tokens

NRP: 1349 Tokens

Okay, so I've got this problem to solve. Let's read it carefully.\n\nThe problem is about Slavic who wants to maximize the product of digits in an array after adding 1 to exactly one of the digits. So, for each test case, I need to find which digit to increment by 1 such that the product becomes as large as possible.\n\nHmm, let's think about the approach. For each digit in the array, I can consider increasing it by 1 and then calculate the product. Then, I pick the maximum among all these possibilities.\n\nWait, but the array could have a zero. Oh right, because if a digit is 9, adding 1 makes it 10. But if a digit is 0, adding 1 makes it 1, which might be better in some cases.\n\nLet me outline the steps:\n\n1. For each test case:\n a. Read the array of digits.\n b. For each digit in the array, create a copy where that digit is increased by 1. But wait, if the digit is 9, adding 1 would make it 10. But since we're treating each digit as a single digit, maybe I should represent them as integers. So, for each element a_i, I'll compute a_i + 1, but only if a_i is 9, it becomes 10. [omitted...]\n\nSo the code structure would be something like:\n\nfor _ in range(t):\n **n = int(input())\n a = list(map(int, input().split()))\n max_prod = 0\n for i in range(n):\n new_a = a.copy()\n new_a[i] += 1\n current_prod = 1\n for num in new_a:\n current_prod *= num\n if current_prod > max_prod:\n max_prod = current_prod\n print(max_prod)**

Remaining: 2026 Tokens

Wait, but wait. What if the original array has a zero, and adding 1 to it? [omitted]\n\nNow, for the code:\n**\nRead t.\n\nFor each test case:\n\n Read n.\n\n Read the list as integers.\n\n Initialize max_prod to zero.\n\n For each i in 0..n-1:\n\n new_a = a.copy()\n\n new_a[i] +=1\n\n compute product of new_a.\n\n if product is larger than current max_prod, update.\n\n print max_prod.**\n\nBut wait, in the case where all digits are zero except one. [omitted]\n\nSo the code:\n\n**nt = int(input())\nfor _ in range(t):\n n = int(input())\n a = list(map(int, input().split()))\n max_product = 0\n for i in range(n):\n new_a = a.copy()\n new_a[i] += 1\n product = 1\n for num in new_a:\n product *= num\n if product > max_product:\n max_product = product\n print(max_product)**\n\nWait, but wait. What if the array is [0, 0, 0], the original product is 0. Adding 1 [omitted]\n\nSo the code should be correct.\n\nThus, the code is as written above.\n

NRP Ratio: 40%