CSES Problem Set

Elevator Rides

Submission details

Task:	Elevator Rides
Sender:	seleneal1996
Submission time:	2021-11-26 06:43:40
Language:	C++17
Status:	READY
Result:	ACCEPTED

Test results

test	verdict	time	
#1	ACCEPTED	0.01 s	»
#2	ACCEPTED	0.01 s	»
#3	ACCEPTED	0.01 s	»
#4	ACCEPTED	0.01 s	»
#5	ACCEPTED	0.01 s	»
#6	ACCEPTED	0.14 s	»
#7	ACCEPTED	0.14 s	»
#8	ACCEPTED	0.14 s	»
#9	ACCEPTED	0.14 s	»
#10	ACCEPTED	0.14 s	»
#11	ACCEPTED	0.14 s	»
#12	ACCEPTED	0.14 s	»
#13	ACCEPTED	0.14 s	»
#14	ACCEPTED	0.15 s	»
#15	ACCEPTED	0.14 s	»
#16	ACCEPTED	0.14 s	»
#17	ACCEPTED	0.14 s	»
#18	ACCEPTED	0.14 s	»
#19	ACCEPTED	0.14 s	»
#20	ACCEPTED	0.15 s	»
#21	ACCEPTED	0.01 s	»
#22	ACCEPTED	0.14 s	»
#23	ACCEPTED	0.13 s	»
#24	ACCEPTED	0.12 s	»
#25	ACCEPTED	0.01 s	»

Dynamic Programming

...

- Money Sums –
- Removal Game –
- Two Sets II –
- Increasing Subsequence –
- Projects –
- Elevator Rides ✓
- Counting Tilings –
- Counting Numbers –

Your submissions

- 2021-11-26 06:43:40 ✓
- 2021-11-26 06:24:17 ×

test	verdict	time	
#26	ACCEPTED	0.13 s	»
#27	ACCEPTED	0.13 s	»
#28	ACCEPTED	0.12 s	»
#29	ACCEPTED	0.14 s	»
#30	ACCEPTED	0.01 s	»
#31	ACCEPTED	0.01 s	»
#32	ACCEPTED	0.12 s	»
#33	ACCEPTED	0.12 s	»
#34	ACCEPTED	0.01 s	»
#35	ACCEPTED	0.12 s	»
#36	ACCEPTED	0.15 s	»
#37	ACCEPTED	0.12 s	»
#38	ACCEPTED	0.01 s	»

Code

```
1  //https://cses.fi/problemset/task/1653
2  #include <bits/stdc++.h>
3  class ElevatorRides{
4  public:
5    void Solve(int n,int k)
6    {
7      int a[n];
8      for (int i = 0; i < n; i++)
9        std::cin>>a[i];
10     std::pair<int, int> dp[1<<n];
11     dp[0] = {0, k+1};
12     for (int s = 1; s < (1<<n); s++)
13     {
14       dp[s] = {25, 0};
15         for (int i = 0; i < n; i++)
16         {
17           if (s>>i&1)
18           {
19             auto [c, w] = dp[s^(1<<i)];
20             if (w + a[i] > k)
21             {
22               c++;
23               w = fmin(a[i], w);
24             }
25             else
26                 w += a[i];
27             dp[s] = min(dp[s], {c, w});
28           }
29         }
30     }
31     std::cout<<dp[(1<<n)-1].first;
32   }
33 };
34
35 int main()
36 {
37   std::ios_base::sync_with_stdio(false);
38   std::cin.tie(0);
39   ElevatorRides S1= ElevatorRides();
40   int n,k;
```

```
41    std::cin>>n>>k;
42    S1.Solve(n,k);
43    return 0;
44  }
```

Share code to others

Test details

Test 1

Verdict: ACCEPTED

input
10 100 36 16 7 33 2 53 25 48 32 11

correct output
3

user output
3

Test 2

Verdict: ACCEPTED

input
10 100 3 32 26 26 64 34 46 70 35 48

correct output
4

user output
4

Test 3

Verdict: ACCEPTED

input
10 100 41 14 19 71 21 45 27 8 52 73

correct output
4

user output
4

Test 4

Verdict: ACCEPTED

input
10 100 20 19 28 56 20 19 63 52 25 60

correct output
4

user output
4

Test 5

Verdict: ACCEPTED

input
10 100 52 28 14 70 27 11 11 40 16 50

correct output
4

user output
4

Test 6

Verdict: ACCEPTED

input
20 1000000000 609930576 743367699 654625611 ...

correct output
14

user output
14

Test 7

Verdict: ACCEPTED

input
20 1000000000 694970558 261292189 582922378 ...

correct output
9

user output
9

Test 8

Verdict: ACCEPTED

input
20 1000000000 188194840 463651706 577460550 ...

correct output
8

user output
8

Test 9

Verdict: ACCEPTED

input
20 1000000000 643426892 4189142 121707902 43...

correct output
6

user output
6

Test 10

Verdict: ACCEPTED

input
20 1000000000 556514452 654521001 282817505 ...

correct output
8

user output
8

Test 11

Verdict: ACCEPTED

input
20 10 2 5 10 4 4 5 4 3 9 2 10 2 7 8 ...

correct output
10

user output
10

Test 12

Verdict: ACCEPTED

input
20 10 9 2 10 2 6 9 6 5 1 6 2 9 3 3 4...

correct output
11

user output
11

Test 13

Verdict: ACCEPTED

input

```
20 10
4 10 1 8 3 3 9 5 5 3 9 8 3 9 2...
```

correct output

```
11
```

user output

```
11
```

Test 14

Verdict: ACCEPTED

input

```
20 10
2 6 5 2 4 3 10 3 1 10 8 6 2 6 ...
```

correct output

```
11
```

user output

```
11
```

Test 15

Verdict: ACCEPTED

input

```
20 10
3 2 1 10 3 3 8 3 4 6 10 4 5 2 ...
```

correct output

```
10
```

user output

```
10
```

Test 16

Verdict: ACCEPTED

input

```
20 10
7 3 9 9 9 10 2 4 3 2 3 3 4 10 ...
```

correct output
11

user output
11

Test 17

Verdict: ACCEPTED

input
20 10 1 3 7 1 1 8 4 3 9 3 3 7 1 4 2 ...

correct output
9

user output
9

Test 18

Verdict: ACCEPTED

input
20 10 4 2 7 10 3 7 9 5 5 10 2 5 2 10...

correct output
11

user output
11

Test 19

Verdict: ACCEPTED

input
20 10 9 3 2 9 7 9 10 5 9 2 5 8 2 3 1...

correct output
12

user output
12

Test 20

Verdict: ACCEPTED

input
20 10 10 6 4 10 9 3 7 9 3 2 2 6 10 2...

correct output
13

user output
13

Test 21

Verdict: ACCEPTED

input
16 1000 589 17 199 306 495 559 14 269 ...

correct output
6

user output
6

Test 22

Verdict: ACCEPTED

input
20 1000000000 609930576 743367699 654625611 ...

correct output
13

user output
13

Test 23

Verdict: ACCEPTED

input
20 100 5 8 10 10 15 16 18 20 25 25 25...

correct output
7

user output
7

Test 24

Verdict: ACCEPTED

input
20 10 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ...

correct output
2

user output
2

Test 25

Verdict: ACCEPTED

input
4 10 4 8 6 1

correct output
2

user output
2

Test 26

Verdict: ACCEPTED

input
20 100 4 8 10 10 15 16 18 20 25 25 25...

correct output
7

user output
7

Test 27

Verdict: ACCEPTED

input
20 100 3 8 10 10 15 16 18 20 25 25 25...

correct output
7

user output
7

Test 28

Verdict: ACCEPTED

input
20 9 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 ...

correct output
6

user output
6

Test 29

Verdict: ACCEPTED

input

```
20 100
41 42 43 44 45 46 47 48 49 59 ...
```

correct output

```
10
```

user output

```
10
```

Test 30

Verdict: ACCEPTED

input

```
1 1
1
```

correct output

```
1
```

user output

```
1
```

Test 31

Verdict: ACCEPTED

input

```
2 2
2 2
```

correct output

```
2
```

user output

```
2
```

Test 32

Verdict: ACCEPTED

input
20 108 1 3 5 6 8 10 11 13 15 16 18 20...

correct output
4

user output
4

Test 33

Verdict: ACCEPTED

input
20 69 10 10 10 10 10 10 10 10 10 10 ...

correct output
4

user output
4

Test 34

Verdict: ACCEPTED

input
14 16 7 13 11 1 7 16 5 3 2 1 12 8 16...

correct output
7

user output
7

Test 35

Verdict: ACCEPTED

input
20 900000000 207900850 208829300 203125674 ...

correct output
6

user output
6

Test 36

Verdict: ACCEPTED

input
20 1000000000 1000000000 1000000000 10000000...

correct output
20

user output
20

Test 37

Verdict: ACCEPTED

input
20 280 56 56 56 56 56 40 40 40 40 40 ...

correct output
3

user output
3

Test 38

Verdict: ACCEPTED

input
1 5 2

correct output

1

user output

1