My approach to the problem:

I had 3 different approaches to the problem

1. The slowest person, Dave goes through the bridge while carrying the torch, and the other cross through the bridge meanwhile.

   The problem with this approach was that, because the problem states minor injuries, so maybe, the people need support and hence in pair. And also, we don't know about the length of the bridge, whether just having the torch at one point be sufficient or not.

2. The fastest person, that is Adam, goes to-and-fro with the torch.

   Doing this I discovered, that the time taken was 65 minutes rather than 60 minutes.

Adam and Bob Go	**10 minutes**
Adam returns	**5 minutes**
Adam and Clair go	**20 minutes**
Adam returns	**5 minutes**
Adam and Dave go	**25 minutes**
(total)	***65 minutes***

3. There are two queues one is for waiting people and the other is for the people who have crossed.

   When a pair crosses the bridge, they migrate from the waiting queue to the reached queue and from the other side of the bridge the first person of the queue comes back with torch and gets appended into the waiting queue. This process continues till the length of the reached queue is 4, that is the total number of people.

   This turned out to be giving the best results, and hence I proceeded with this

Adam and Bob go	**10 minutes**
Adam returns	**5 minutes**
Clair and Dave go	**25 minutes**
Bob returns	**10 minutes**

Adam and Bob go	**10 minutes**
(total)	***60 minutes***

Reflection to my code

My code contains two arrays, ***arr1*** and ***arr2*** which act as two queues.

The first one is for the *waiting list* and the other is for the *reached list.*

There are 4 flag variables which denote the start and end position of the queues. There is a ***time*** variable which keeps into account the amount of time taken.

All the variables and arrays are *global.*

There are 3 functions, namely, ***decisionFunc(), toReached(), backWait() .***

decisionFunc() is used to send the first pair from *waiting*(arr1) to ***toReached().***

toReached() is used to insert the element received, into the *reached queue (arr2)*. It then sends the first element of Reached queue to ***backWait()*** function. This function also checks whether the length of reached queue (arr2) is 4 or not, if the length is 4, then no element is sent to ***backWait().***

backWait() is used to insert the variable received in the waiting (arr1) queue.

Further Improvements:-

1. The arrays should be dynamic, so that, we can take as many people as we want and not just 4.
2. There should be user input.

3. Instead of using the true value *4*, using a more generalized term like *n,* which will have the total number of people.