

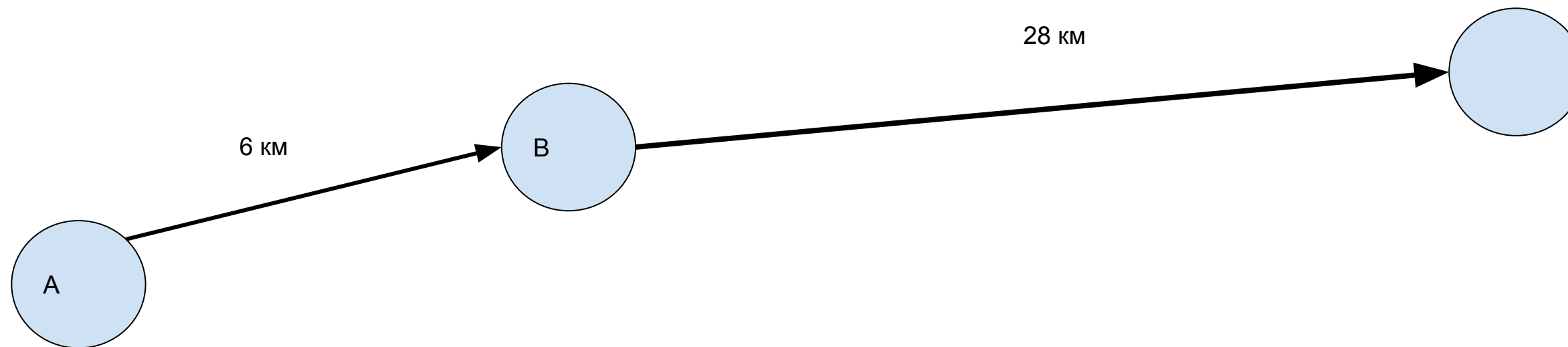
# Perfect number

$$6 = 1 + 2 + 3$$

$$28 = 1 + 2 + 4 + 7 + 14$$

51) 900 000 000 digits 2019

10000000\$



```
void perfect(int number)
{
}
```

number=100  
6  
28

number=1000  
1)6  
2)28  
3)496

```
void perfect(int number) {
    int sum;
    for (int i = 1; i <= number; i++) { // starts loop to check all numbers less
    than "entered" number
        sum = 0;
        for (int j = 1; j < i; j++) { // starts loop for all divisors of each
        number?
            if (i % j == 0) { // checking if j is a divisor of i
                sum = sum + j; // adding up divisors of number
            }
        }
        if (sum == i) {
            std::cout << "This is a perfect number: " << i << std::endl;
        }
    }
}
```

$$2^{(p-1)} * (2^p - 1)$$

$(2^p - 1)$  - prime number

- p=2  $2^{(p-1)} * (2^p - 1) = 2^{(2-1)} * (2^2 - 1) = 2 * 3 = 6$
- p=3  $2^{(p-1)} * (2^p - 1) = 2^{(3-1)} * (2^3 - 1) = 4 * 7 = 28$
- p=4  $2^{(p-1)} * (2^p - 1) = 2^{(4-1)} * (2^4 - 1) = 8 * 15 = 120$
- p=5  $2^{(p-1)} * (2^p - 1) = 2^{(5-1)} * (2^5 - 1) = 16 * 31 = 496$

- 1) how many is the problem
- 2) if exist any odd perfect number