

11. What is the greatest product of four adjacent numbers in the same direction (vertical, horizontal, diagonal) in a given  $20 \times 20$  grid?

ex. 

08	02	22	97	38	...
49	49	99	40	17	...
81	49	31	73	55	...
52	20	95	23	04	...
:	:	:	:	:	:

 ... return the product of these four

Some diagonal edge cases

B			A		C		D
	B	A				C	D
	A	B				D	C
A			B		D		C
					E		F
						E	F
						F	E
					F		E

call the square  $s$  and index  $i$  -

line B:  $s[i] + s[2 \cdot i + 1] + s[3 \cdot i + 2] + s[4 \cdot i + 3]$   
etc.

```

n = '08 02 22 ...' \
    '49 49 99 ...' \
    :   :   :   \
    '01 70 54 ...'

```

```

n = n.split
def se-product(number, width, digits)
  substr_list = []
  #horizontal
  index = 0
  while index < n.length - digits
    substr_list << n[index..(index+digits-1)]
    index += 1
  end
  #vertical
  index = 0
  while index < number.length - (width*3)
    substr_list << [number[index],
                    number[index+width],
                    number[index+2*width],
                    number[index+3*width]]
    index += 1
  end
  #UR-LL diagonal
  \ \

```

```

index = 0
while index < number.length - (width * 3)
  index += 3 if (index % width).zero?
  substr_list << [number[index],
                  number[index + width - 1],
                  number[index + (2 * width) - 2],
                  number[index + (3 * width) - 3]]
  index += 1
end
# UL-LR diagonal
index = 0
while index < number.length - (width * 3) - 3
  index += 3 if ((index + 3) % width).zero?
  substr_list << [number[index],
                  number[index + width + 1],
                  number[index + (2 * width) + 2],
                  number[index + (3 * width) + 3]]
  index += 1
end
substr_list.map! do |str| # replace each
  str.map do |digits| # subarray with
    digits.to_i.reduce(:*) # the product
  end # of the nums.
end # it contains
substr_list.max
end

```

