1. An oil spill has fouled 200 miles of Pacific shoreline. The oil company responsible has been given 14 days to clean up the shoreline, after which a fine will be levied in the amount of $10,000/day. The local cleanup crew can scrub five miles of beach per week at a cost of $500/day. Additional crews can be brought in at a cost of $18,000 plus $800/day for each crew.

(i) Let $T(n)$ be the number of days needed to clean the shoreline expressed as a function n of the number of additional crews hired. Write an expression for $T(n)$.

In [1]:
```
using Symbolics
D(f,x)=expand_derivatives(Differential(x)(f))
@variables n
```

Out[1]:
```
1-element Vector{Num}:
 n
```

Given n additional crews, the total number of crews working is $n+1$. We assume each crew can scrub five miles of beach per week, so with n additional crews $5(n+1)$ miles of beach per week can be cleaned. Since there are 7 days in a week, it follows that the cleanup rate $R(n)$ is miles/day is given by

$$R(n) = 5(n+1)\,\frac{\text{miles}}{\text{week}} \times \frac{1}{7}\,\frac{\text{week}}{\text{day}} = \frac{5}{7}(n+1)\,\frac{\text{miles}}{\text{day}}.$$

Since there are 200 miles that need to be cleaned then the total number of days needed to clean the spill is

$$T(n) = \frac{200}{R(n)} = \frac{280}{n+1}\ \text{days}.$$

In [2]:
```
T(n)=7*200/(5*(n+1))
T(n)
```

Out[2]:
```
280 / (1 + n)
```

(ii) Let the set $I_1 = \{\, n : T(n) \le 14 \,\}$ be the choices of n which avoid a fine and let $I_2 = \{n : T(n) > 14\}$ be the choices which lead to a fine. Find I_1 and I_2 .

To find I_1 write

$$\frac{280}{n+1} \le 14 \qquad \text{so that} \qquad n+1 \ge \frac{280}{14} = 20.$$

It follows that $n \ge 19$ and so $I_1 = \{\, 19, 20, 21, \ldots, \,\}$.

To find I_2 note I_1 and I_2 are disjoint and $I_1 \cup I_2 = \{0\} \cup \mathbf{N}$.

Therefore $I_2 = \{0\} \cup \mathbf{N} \setminus I_1 = \{\, 0, 1, \ldots, 18 \,\}$.

(iii) The total cost of the cleanup is

$$C(n) = \underbrace{(500 + 800n)T(n)}_{\text{daily costs}} + \underbrace{18000n}_{\text{crew setup}} + \underbrace{F(n)}_{\text{fine}}$$

where the fine is given by

$$F(n) = \begin{cases} 0 & \text{for } n \in I_1 \\ 10000\big(T(n) - 14\big) & \text{for } n \in I_2. \end{cases}$$

Plot the points $\big(n, C(n)\big)$ for the $n = 0, 1, \ldots, 20$.

Hint: In Julia one can define `F(n)=10000*max(T(n)-14,0)` and use the `scatter` command in the `Plots` library for plotting points.

In [3]:
```
F(n)=10000*max(T(n)-14,0)
F(n)
```

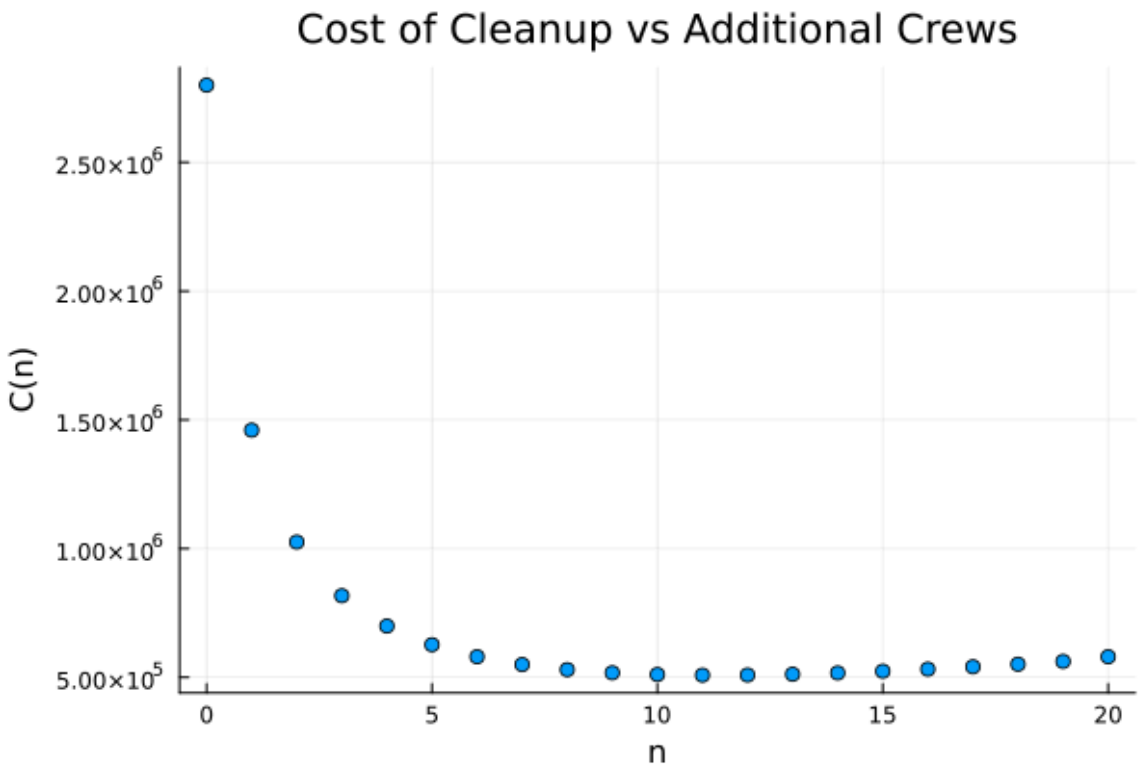
Out[3]:
```
10000max(-14 + 280 / (1 + n), 0)
```

In [4]:
```
C(n)=(500+800n)*T(n)+18000*n+F(n)
C(n)
```

Out[4]:
```
18000n + 10000max(-14 + 280 / (1 + n), 0) + (280(500 + 800n)) / (1 + n)
```

In [5]:
```
using Plots
scatter(n->n,n->C(n),0:20,
    xlabel="n",ylabel="C(n)",title="Cost of Cleanup vs Additional Crews",leg
```

Out[5]:



(iv) Compute $C(n)$ for $n = 0, \ldots, 20$ and determine the optimal value for n.

Hint: $n = 11$ is optimal but include a table of all 21 values of n and $C(n)$.

In [6]:
```
# Include the time T(n) and fine F(n) for better understanding
A=vcat([[n T(n) F(n) C(n)] for n=0:20]...)
```

Out[6]:
```
21×4 Matrix{Float64}:
  0.0  280.0          2.66e6          2.8e6
  1.0  140.0          1.26e6          1.46e6
  2.0   93.3333       7.93333e5       1.02533e6
  3.0   70.0      560000.0         817000.0
  4.0   56.0      420000.0         699200.0
  5.0   46.6667       3.26667e5       6.26667e5
  6.0   40.0      260000.0         580000.0
  7.0   35.0      210000.0         549500.0
  8.0   31.1111       1.71111e5       5.29778e5
  9.0   28.0      140000.0         517600.0
 10.0   25.4545       1.14545e5       5.10909e5
 11.0   23.3333   93333.3             5.08333e5
 12.0   21.5385   75384.6             5.08923e5
 13.0   20.0      60000.0          512000.0
 14.0   18.6667   46666.7             5.17067e5
 15.0   17.5      35000.0          523750.0
 16.0   16.4706   24705.9             5.31765e5
 17.0   15.5556   15555.6             5.40889e5
 18.0   14.7368    7368.42            5.50947e5
 19.0   14.0          0.0          561800.0
 20.0   13.3333       0.0          580000.0
```

In [7]:
```
# Make a nicer table (optional)
using PrettyTables
pretty_table(A,
    formatters=[fmt__printf("%3d",[1]),fmt__printf("%.2f",[2,3,4])],
    column_labels=["n","T(n)","F(n)","C(n)"])
```

n	T(n)	F(n)	C(n)
0	280.00	2660000.00	2800000.00
1	140.00	1260000.00	1460000.00
2	93.33	793333.33	1025333.33
3	70.00	560000.00	817000.00
4	56.00	420000.00	699200.00
5	46.67	326666.67	626666.67
6	40.00	260000.00	580000.00
7	35.00	210000.00	549500.00
8	31.11	171111.11	529777.78
9	28.00	140000.00	517600.00
10	25.45	114545.45	510909.09
11	23.33	93333.33	508333.33
12	21.54	75384.62	508923.08
13	20.00	60000.00	512000.00
14	18.67	46666.67	517066.67
15	17.50	35000.00	523750.00
16	16.47	24705.88	531764.71
17	15.56	15555.56	540888.89
18	14.74	7368.42	550947.37
19	14.00	0.00	561800.00
20	13.33	0.00	580000.00

In [8]:
```
nopt=A[argmin(A[:,4])]
```

Out[8]:
```
11.0
```

The optimal number of additional crews is $n = 11$.

The company has filed an appeal on grounds that the fine of $10,000/day for missing the deadline is excessive. Let M be the daily fine. What is the least value for M that ensures the cost-optimized value of $T(n)$ meets the 14 day deadline? Is M larger or smaller than $10,000/day?

In [9]:
```
F(nopt)
```

Out[9]:
```
93333.33333333333
```

In [10]:
```
T(nopt)
```

Out[10]:
```
23.333333333333332
```

The fine and time of cleanup at the optimal value of n is

$$F(n_{\text{opt}}) \approx 93333.33 \qquad \text{and} \qquad T(n_{\text{opt}}) \approx 23.33.$$

Thus, the fine of $10000/day was not enough to avoid missing the deadline.

For no fine we have $n \in I_1$, thus the fewest additional work crews needed is 19 to avoid the fine. Let M be the daily fine

In [11]:
```
@variables M
F(n)=M*max(T(n)-14,0)
C(n)
```

Out[11]:
```
18000n + (280(500 + 800n)) / (1 + n) + M*max(-14 + 280 / (1 + n), 0)
```

What we now seek is the minimum value of M when $C(19) < C(18)$. This will ensure that the cost-optimized number of work crews will be exactly $n = 19$ which is the minimum needed to meet the deadline.

In [12]:
```
delta=C(19)-C(18)
```

Out[12]:
```
18221.05263157899 - 0.7368421052631575M
```

In [13]:
```
Mn=Symbolics.solve_for(delta,M)
Mmin=eval(Symbolics.toexpr(Mn))
```

Out[13]:
```
24728.5714285715
```

Rounding up indicates that $M = 24728.58$ will ensure the cost-optimized cleanup meets the deadline.

To verify that $24728.58/day is a sufficient fine check the cost optimized number of additional work crews.

In [14]:
```
M=24728.58
A=vcat([[n T(n) F(n) C(n)] for n=10:30]...)
pretty_table(A,
    formatters=[fmt__printf("%3d",[1]),fmt__printf("%.2f",[2,3,4])],
    column_labels=["n","T(n)","F(n)","C(n)"])
```

n	T(n)	F(n)	C(n)
10	25.45	283254.64	679618.28
11	23.33	230800.08	645800.08
12	21.54	186415.45	619953.91
13	20.00	148371.48	600371.48
14	18.67	115400.04	585800.04
15	17.50	86550.03	575300.03
16	16.47	61094.14	568152.96
17	15.56	38466.68	563800.01
18	14.74	18221.06	561800.01
19	14.00	0.00	561800.00
20	13.33	0.00	580000.00
21	12.73	0.00	598181.82
22	12.17	0.00	616347.83
23	11.67	0.00	634500.00
24	11.20	0.00	652640.00
25	10.77	0.00	670769.23
26	10.37	0.00	688888.89
27	10.00	0.00	707000.00
28	9.66	0.00	725103.45
29	9.33	0.00	743200.00
30	9.03	0.00	761290.32

In [15]:
```
nopt=A[argmin(A[:,4])]
```

Out[15]: 19.0

This confirms that $24728.58/day is a sufficient fine check the cost optimized number of additional work crews.

An alternative way to work this problem which gives a reasonable approximation for M is to assume n is continuous and then differentiate.

Consider the cost when $n \leq 19$ given by

$$C_2(n) = (500 + 800n)T(n) + 18000n + M(T(n) - 14).$$

and solve for M such that $C_2'(19) = 0$.

Note since we have assumed n to be a continuous variable this isn't actually the smallest value of M that guarantees the deadline will be met but only a reasonable approximation that is larger than the one obtained earlier.

In [16]:
```
@variables M
C2(n)=(500+800n)*T(n)+18000*n+M*(T(n)-14)
C2(n)
```

Out[16]: M*(-14 + 280 / (1 + n)) + 18000n + (280(500 + 800n)) / (1 + n)

In [17]:
```
dC2n=simplify(D(C2(n),n))
```

Out[17]:
```
(-102000 + 280M - 36000n - 18000(n^2)) / (-1 - 2n - (n^2))
```

In [18]:
```
dC2s="dC2(n)="*string(dC2n)
eval(Meta.parse(dC2s))
dC2(n)
```

Out[18]:
```
(-102000 + 280M - 36000n - 18000(n^2)) / (-1 - 2n - (n^2))
```

In [19]:
```
Mn=Symbolics.solve_for(dC2(19),M)
```

Out[19]:
```
182100//7
```

In [20]:
```
Mapp=Float64(Symbolics.toexpr(Mn))
```

Out[20]:
```
26014.285714285714
```

In [21]:
```
M=Mapp
A=vcat([[n T(n) F(n) C(n)] for n=10:30]...)
pretty_table(A,
    formatters=[fmt__printf("%3d",[1]),fmt__printf("%.2f",[2,3,4])],
    column_labels=["n","T(n)","F(n)","C(n)"])
```

n	T(n)	F(n)	C(n)
10	25.45	297981.82	694345.45
11	23.33	242800.00	657800.00
12	21.54	196107.69	629646.15
13	20.00	156085.71	608085.71
14	18.67	121400.00	591800.00
15	17.50	91050.00	579800.00
16	16.47	64270.59	571329.41
17	15.56	40466.67	565800.00
18	14.74	19168.42	562747.37
19	14.00	0.00	561800.00
20	13.33	0.00	580000.00
21	12.73	0.00	598181.82
22	12.17	0.00	616347.83
23	11.67	0.00	634500.00
24	11.20	0.00	652640.00
25	10.77	0.00	670769.23
26	10.37	0.00	688888.89
27	10.00	0.00	707000.00
28	9.66	0.00	725103.45
29	9.33	0.00	743200.00
30	9.03	0.00	761290.32

In [22]:
```
nopt=A[argmin(A[:,4])]
```

Out[22]:
```
19.0
```

Again the cost optimized solution meets the exact deadline for the cleanup. However, M is slightly bigger than it needs to be. Note that I have included this

possible way of working the problem to have an answer to compare in anticipation that some people may have done it this way.

In [ ]: