COMP9414 Artificial Intelligence

Assignment 1: Constraint Satisfaction Search

@Authors: **Wayne Wobcke, Alfred Krzywicki, Stefano Mezza**

Due Date: Week 5, Friday, October 17, 5.00pm

Objective

This assignment concerns developing optimal solutions to a scheduling problem inspired by the scenario of a manufacturing plant that has to fulfil multiple customer orders with varying deadlines, but where there may be constraints on tasks and on relationships between tasks. Any number of tasks can be scheduled at the same time, but it is possible that some tasks cannot be finished before their deadline. A task finishing late is acceptable, however incurs a cost, which for this assignment is a simple (dollar) amount per hour that the task is late.

A *fuzzy scheduling* problem in this scenario is simplified by ignoring customer orders and having just one machine and a number of *tasks*, each with a fixed duration in hours. Each task must start and finish on the same day, within working hours (9am to 5pm). In addition, there can be *constraints*, both on single tasks and between two tasks. One type of constraint is that a task can have a deadline, which can be "hard" (the deadline must be met in any valid schedule) or "soft" (the task may be finished late – though still at or before 5pm – but with a "cost" per hour for missing the deadline). The aim is to develop an overall schedule for all the tasks (in a single week) that minimizes the total cost of all the tasks that finish late, provided that all the hard constraints on tasks are satisfied.

More technically, this assignment is an example of a *constraint optimization problem* (or *constrained optimization problem*), a problem that has constraints like a standard Constraint Satisfaction Problem (CSP), but also a *cost* associated with each solution. For this assignment, we will use a *greedy* algorithm to find optimal solutions to fuzzy scheduling problems that are specified as text strings. However, unlike the greedy search algorithm described in the lectures on search, this greedy algorithm has the property that it is guaranteed to find an optimal solution for any problem (if a solution exists).

The assignment will use the AIPython code of Poole & Mackworth. You are given code to translate fuzzy scheduling problems specified as text strings into CSPs with a cost, and you are given code for several constraint solving algorithms – based on domain splitting and arc consistency, and based on depth-first search. The assignment will be to implement some missing procedures and to analyse the performance of the constraint solving methods, both analytically and experimentally.

Submission Instructions

- This is an individual assignment.
- Write your answers in **this** notebook and submit **this** notebook on Moodle under **Assignment 1**.
- Name your submission `<zid>-<firstname>-<lastname>.ipynb` where `<firstname>-<lastname>` is your **real** (not Moodle) name.
- Make sure you set up AIPython (as done below) so the code can be run on either CSE machines or a marker's own machine.
- Do not submit any AIPython code. Hence do not change any AIPython code to make your code run.
- Make sure your notebook runs cleanly (restart the kernel, clear all outputs and run each cell to check).
- After checking that your notebook runs cleanly, run all cells and submit the notebook **with** the outputs included (do not submit the empty version).
- Make sure images (for plots/graphs) are **included** in the notebook you submit (sometimes images are saved on your machine but are not in the notebook).
- Do not modify the existing code in this notebook except to answer the questions. Marks will be given as and where indicated.
- If you want to submit additional code (e.g. for generating plots), add that at the end of the notebook.
- **Important: Do not distribute any of this code on the Internet. This includes ChatGPT. Do not put this assignment into any LLM.**

Late Penalties

Standard UNSW late penalties apply (5% of the value of the assignment per day or part day late).

Note: Unlike the CSE systems, there is no grace period on Moodle. The due date and time is 5pm **precisely** on Friday October 17.

Important: You can submit as many times as you want before the due date, but if you do submit before the due date, you cannot submit on Moodle after the due date. If you do not submit before the due date, you can submit on Moodle after the due date.

Plagiarism

Remember that ALL work submitted for this assignment must be your own work and no sharing or copying of code or answers is allowed. You may discuss the assignment with other students but must not collaborate on developing answers to the questions. You

may use code from the Internet only with suitable attribution of the source. You may not use ChatGPT or any similar software to generate any part of your explanations, evaluations or code. Do not use public code repositories on sites such as github or file sharing sites such as Google Drive to save any part of your work – make sure your code repository or cloud storage is private and do not share any links. This also applies after you have finished the course, as we do not want next year's students accessing your solution, and plagiarism penalties can still apply after the course has finished.

All submitted assignments will be run through plagiarism detection software to detect similarities to other submissions, including from past years. You should **carefully** read the UNSW policy on academic integrity and plagiarism (linked from the course web page), noting, in particular, that collusion (working together on an assignment, or sharing parts of assignment solutions) is a form of plagiarism.

Finally, do not use any contract cheating "academies" or online "tutoring" services. This counts as serious misconduct with heavy penalties up to automatic failure of the course with 0 marks, and expulsion from the university for repeat offenders.

Fuzzy Scheduling

A CSP for this assignment is a set of variables representing tasks, binary constraints on pairs of tasks, and unary constraints (hard or soft) on tasks. The domains are all the working hours in one week, and a task duration is in hours. Days are represented (in the input and output) as strings 'mon', 'tue', 'wed', 'thu' and 'fri', and times are represented as strings '9am', '10am', '11am', '12pm', '1pm', '2pm', '3pm', '4pm' and '5pm'. The only possible values for the start and end times of a task are combinations of a day and times, e.g. 'mon 9am'. Each task name is a string (with no spaces), and the only soft constraints are the soft deadline constraints.

There are three types of constraint:

- **Binary Constraints:** These specify a hard requirement for the relationship between two tasks.
- **Hard Domain Constraints:** These specify hard requirements for the tasks themselves.
- **Soft Deadline Constraints:** These constraints specify that a task may finish late, but with a given cost.

Each soft constraint has a function defining the *cost* associated with violating the preference, that the constraint solver must minimize, while respecting all the hard constraints. The *cost* of a solution is simply the sum of the costs for the soft constraints that the solution violates (and is always a non-negative integer).

This is the list of possible constraints for a fuzzy scheduling problem (comments below are for explanation and do **not** appear in the input specification; however, the code we supply *should* work with comments that take up a full line):

```
# binary constraints
constraint, ⟨t1⟩ before ⟨t2⟩           # t1 ends when or before
t2 starts
constraint, ⟨t1⟩ after ⟨t2⟩            # t1 starts after or when
t2 ends
constraint, ⟨t1⟩ same-day ⟨t2⟩         # t1 and t2 are scheduled
on the same day
constraint, ⟨t1⟩ starts-at ⟨t2⟩        # t1 starts exactly when
t2 ends

# hard domain constraints
domain, ⟨t⟩, ⟨day⟩, hard                                    # t
starts on given day at any time
domain, ⟨t⟩, ⟨time⟩, hard                                   # t
starts at given time on any day
domain, ⟨t⟩, starts-before ⟨day⟩ ⟨time⟩, hard               # t
starts at or before day, time
domain, ⟨t⟩, starts-after ⟨day⟩ ⟨time⟩, hard                # t
starts at or after day, time
domain, ⟨t⟩, ends-before ⟨day⟩ ⟨time⟩, hard                 # t
ends at or before day, time
domain, ⟨t⟩, ends-after ⟨day⟩ ⟨time⟩, hard                  # t
starts at or after day, time
domain, ⟨t⟩, starts-in ⟨day1⟩ ⟨time1⟩-⟨day2⟩ ⟨time2⟩, hard  # day-
time range for start time; includes day1, time1 and day2, time2
domain, ⟨t⟩, ends-in ⟨day1⟩ ⟨time1⟩-⟨day2⟩ ⟨time2⟩, hard    # day-
time range for end time; includes day1, time1 and day2, time2
domain, ⟨t⟩, starts-before ⟨time⟩, hard                     # t
starts at or before time on any day
domain, ⟨t⟩, ends-before ⟨time⟩, hard                       # t
ends at or before time on any day
domain, ⟨t⟩, starts-after ⟨time⟩, hard                      # t
starts at or after time on any day
domain, ⟨t⟩, ends-after ⟨time⟩, hard                        # t
ends at or after time on any day

# soft deadline constraint
domain, ⟨t⟩, ends-by ⟨day⟩ ⟨time⟩ ⟨cost⟩, soft          # cost per
hour of missing deadline
```

The input specification will consist of several “blocks”, listing the tasks, binary constraints, hard unary constraints and soft deadline constraints for the given problem. A “declaration” of each task will be included before it is used in a constraint. A sample input specification is as follows. Comments are for explanation and do **not** have to be included in the input.

```
# two tasks with two binary constraints and soft deadlines
task, t1 3
task, t2 4
# two binary constraints
constraint, t1 before t2
constraint, t1 same-day t2
# domain constraint
domain, t2 mon
```

```
    # soft deadline constraints
    domain, t1 ends-by mon 3pm 10
    domain, t2 ends-by mon 3pm 10
```

Preparation

1. Set up AIPython

You will need AIPython for this assignment. To find the aipython files, the aipython directory has to be added to the Python path.

Do this temporarily, as done here, so we can find AIPython and run your code (you will not submit any AIPython code).

You can add either the full path (using `os.path.abspath`), or as in the code below, the relative path.

In [2]:
```
import sys
sys.path.append('aipython') # change to your directory
sys.path # check that aipython is now on the path
```

Out[2]:
```
['/opt/anaconda3/envs/comp9517/lib/python310.zip',
 '/opt/anaconda3/envs/comp9517/lib/python3.10',
 '/opt/anaconda3/envs/comp9517/lib/python3.10/lib-dynload',
 '',
 '/opt/anaconda3/envs/comp9517/lib/python3.10/site-packages',
 '/opt/anaconda3/envs/comp9517/lib/python3.10/site-packages/setuptools/_vendo
r',
 'aipython']
```

2. Representation of Day Times

Input and output are day time strings such as 'mon 10am' or a range of day time strings such as 'mon 10am-mon 4pm'.

The CSP will represent these as integer hour numbers in the week, ranging from 0 to 39.

The following code handles the conversion between day time strings and hour numbers.

In [3]:
```
# -*- coding: utf-8 -*-

""" day_time string format is a day plus time, e.g. Mon 10am, Tue 4pm, or just T
    if only day or time, returns day number or hour number only
    day_time strings are converted to and from integer hours in the week from 0
"""
class Day_Time():
    num_hours_in_day = 8
    num_days_in_week = 5

    def __init__(self):
        self.day_names = ['mon','tue','wed','thu','fri']
        self.time_names = ['9am','10am','11am','12pm','1pm','2pm','3pm','4pm']

    def string_to_week_hour_number(self, day_time_str):
```

```
        """ convert a single day_time into an integer hour in the week """
        value = None
        value_type = None
        day_time_list = day_time_str.split()
        if len(day_time_list) == 1:
            str1 = day_time_list[0].strip()
            if str1 in self.time_names: # this is a time
                value = self.time_names.index(str1)
                value_type = 'hour_number'
            else:
                value = self.day_names.index(str1) # this is a day
                value_type = 'day_number'
            # if not day or time, throw an exception
        else:
            value = self.day_names.index(day_time_list[0].strip())*self.num_hour
                + self.time_names.index(day_time_list[1].strip())
            value_type = 'week_hour_number'
        return (value_type, value)

    def string_to_number_set(self, day_time_list_str):
        """ convert a list of day-times or ranges 'Mon 9am, Tue 9am-Tue 4pm' int
            e.g. 'mon 9am-1pm, mon 4pm' -> [0,1,2,3,4,7]
        """
        number_set = set()
        type1 = None
        for str1 in day_time_list_str.lower().split(','):
            if str1.find('-') > 0:
                # day time range
                type1, v1 = self.string_to_week_hour_number(str1.split('-')[0].s
                type2, v2 = self.string_to_week_hour_number(str1.split('-')[1].s
                if type1 != type2: return None # error, types in range spec are
                number_set.update({n for n in range(v1, v2+1)})
            else:
                # single day time
                type2, value2 = self.string_to_week_hour_number(str1)
                if type1 != None and type1 != type2: return None # error: type i
                type1 = type2
                number_set.update({value2})
        return (type1, number_set)

    # convert integer hour in week to day time string
    def week_hour_number_to_day_time(self, week_hour_number):
        hour = self.day_hour_number(week_hour_number)
        day = self.day_number(week_hour_number)
        return self.day_names[day]+' '+self.time_names[hour]

    # convert integer hour in week to integer day and integer time in day
    def hour_day_split(self, week_hour_number):
        return (self.day_hour_number(week_hour_number), self.day_number(week_hou

    # convert integer hour in week to integer day in week
    def day_number(self, week_hour_number):
        return int(week_hour_number / self.num_hours_in_day)

    # convert integer hour in week to integer time in day
    def day_hour_number(self, week_hour_number):
        return week_hour_number % self.num_hours_in_day

    def __repr__(self):
        day_hour_number = self.week_hour_number % self.num_hours_in_day
```

```
        day_number = int(self.week_hour_number / self.num_hours_in_day)
        return self.day_names[day_number]+' '+self.time_names[day_hour_number]
```

3. Constraint Satisfaction Problems with Costs over Tasks with Durations

Since AI Python does not provide the CSP class with an explicit cost, we implement our own class that extends `CSP`.

We also store the cost functions and the durations of all tasks explicitly in the CSP.

The durations of the tasks are used in the `hold` function to evaluate constraints.

In [4]:
```
from cspProblem import CSP, Constraint

# We need to override Constraint, because tasks have durations
class Task_Constraint(Constraint):
    """A Task_Constraint consists of
    * scope: a tuple of variables
    * spec: text description of the constraint used in debugging
    * condition: a function that can applied to a tuple of values for the variab
    * durations: durations of all tasks
    * func_key: index to the function used to evaluate the constraint
    """
    def __init__(self, scope, spec, condition, durations, func_key):
        super().__init__(scope, condition, spec)
        self.scope = scope
        self.condition = condition
        self.durations = durations
        self.func_key = func_key

    def holds(self, assignment):
        """returns the value of Constraint con evaluated in assignment.

        precondition: all variables are assigned in assignment

        CSP has only binary constraints
        condition is in the form week_hour_number1, week_hour_number2
        add task durations as appropriate to evaluate condition
        """
        if self.func_key == 'before':
            # t1 ends before t2 starts, so we need add duration to t1 assignment
            ass0 = assignment[self.scope[0]] + self.durations[self.scope[0]]
            ass1 = assignment[self.scope[1]]
        elif self.func_key == 'after':
            # t2 ends before t1 starts so we need add duration to t2 assignment
            ass0 = assignment[self.scope[0]]
            ass1 = assignment[self.scope[1]] + self.durations[self.scope[1]]
        elif self.func_key == 'starts-at':
            # t1 starts exactly when t2 ends, so we need add duration to t2 assi
            ass0 = assignment[self.scope[0]]
            ass1 = assignment[self.scope[1]] + self.durations[self.scope[1]]
        else:
            return self.condition(*tuple(assignment[v] for v in self.scope))
        # condition here comes from get_binary_constraint
        return self.condition(*tuple([ass0, ass1]))

# implement nodes as CSP problems with cost functions
```

```
class CSP_with_Cost(CSP):
    """ cost_functions maps a CSP var, here a task name, to a list of functions
    def __init__(self, domains, durations, constraints, cost_functions, soft_day
        self.domains = domains
        self.variables = self.domains.keys()
        super().__init__("title of csp", self.variables, constraints)
        self.durations = durations
        self.cost_functions = cost_functions
        self.soft_day_time = soft_day_time
        self.soft_costs = soft_costs
        self.cost = self.calculate_cost()

    # specific to fuzzy scheduling CSP problems
    def calculate_cost(self):
        """ this is really a function f = path cost + heuristic to be used by th
        cost = 0
        for var in self.variables:
            dom = self.domains[var]
            is_cost = self.cost_functions[var][0]
            if is_cost.__name__ == 'no_cost':
                continue
            duration = self.durations[var]
            soft_time = self.soft_day_time[var]
            soft_cost = self.soft_costs[var]
            min_cost = None
            for i in dom:
                penalty = is_cost(i, soft_time, duration, soft_cost)
                if min_cost is None or penalty < min_cost:
                    min_cost = penalty
            if min_cost is None:
                min_cost = 0
            cost += min_cost
        return cost

    def __repr__(self):
        """ string representation of an arc"""
        return "CSP_with_Cost("+str(list(self.domains.keys()))+':'+str(self.cost
```

This formulates a solver for a CSP with cost as a search problem, using domain splitting with arc consistency to define the successors of a node.

In [5]:

```
from cspConsistency import Con_solver, select, partition_domain
from searchProblem import Arc, Search_problem
from operator import eq, le, ge

# rewrites rather than extends Search_with_AC_from_CSP
class Search_with_AC_from_Cost_CSP(Search_problem):
    """ A search problem with domain splitting and arc consistency """
    def __init__(self, csp):
        self.cons = Con_solver(csp) # copy of the CSP with access to arc consist
        self.domains = self.cons.make_arc_consistent(csp.domains)
        self.constraints = csp.constraints
        self.cost_functions = csp.cost_functions
        self.durations = csp.durations
        self.soft_day_time = csp.soft_day_time
        self.soft_costs = csp.soft_costs
        csp.domains = self.domains # after arc consistency
        self.csp = csp
```

```
    def is_goal(self, node):
        """ node is a goal if all domains have exactly 1 element """
        return all(len(node.domains[var]) == 1 for var in node.domains)

    def start_node(self):
        return CSP_with_Cost(self.domains, self.durations, self.constraints,
                             self.cost_functions, self.soft_day_time, self.soft_

    def neighbors(self, node):
        """returns the neighboring nodes of node.
        """
        neighs = []
        var = select(x for x in node.domains if len(node.domains[x]) > 1) # chos
        if var:
            dom1, dom2 = partition_domain(node.domains[var])
            self.display(2, "Splitting", var, "into", dom1, "and", dom2)
            to_do = self.cons.new_to_do(var, None)
            for dom in [dom1,dom2]:
                newdoms = node.domains | {var: dom} # overwrite domain of var wi
                cons_doms = self.cons.make_arc_consistent(newdoms, to_do)
                if all(len(cons_doms[v]) > 0 for v in cons_doms):
                    # all domains are non-empty
                    # make new CSP_with_Cost node to continue the search
                    csp_node = CSP_with_Cost(cons_doms, self.durations, self.con
                             self.cost_functions, self.soft_day_time, self.soft_
                    neighs.append(Arc(node, csp_node))
                else:
                    self.display(2,"...",var,"in",dom,"has no solution")
        return neighs

    def heuristic(self, n):
        return n.cost
```

4. Fuzzy Scheduling Constraint Satisfaction Problems

The following code sets up a CSP problem from a given specification.

Hard (unary) domain constraints are applied to reduce the domains of the variables before the constraint solver runs.

In [6]:
```
# domain specific CSP builder for week schedule
class CSP_builder():
    # list of text lines without comments and empty lines
    _, default_domain = Day_Time().string_to_number_set('mon 9am-fri 4pm') # sho

    # hard unary constraints: domain is a list of values, params is a single val
    # starts-before, ends-before (for starts-before duration should be 0)
    # vals in domain are actual task start/end date/time, so must be val <= what
    def apply_before(self, param_type, params, duration, domain):
        domain_orig = domain.copy()
        param_val = params.pop()
        for val in domain_orig: # val is week_hour_number
            val1 = val + duration
            h, d = Day_Time().hour_day_split(val1)
            if param_type == 'hour_number' and h > param_val:
                if val in domain: domain.remove(val)
            if param_type == 'day_number' and d > param_val:
                if val in domain: domain.remove(val)
```

```
            if param_type == 'week_hour_number' and val1 > param_val:
                if val in domain: domain.remove(val)
        return domain

    def apply_after(self, param_type, params, duration, domain):
        domain_orig = domain.copy()
        param_val = params.pop()
        for val in domain_orig: # val is week_hour_number
            val1 = val + duration
            h, d = Day_Time().hour_day_split(val1)
            if param_type == 'hour_number' and h < param_val:
                if val in domain: domain.remove(val)
            if param_type == 'day_number' and d < param_val:
                if val in domain: domain.remove(val)
            if param_type == 'week_hour_number' and val1 < param_val:
                if val in domain: domain.remove(val)
        return domain

    # day time range only
    # includes starts-in, ends-in
    # duration is 0 for starts-in, task duration for ends-in
    def apply_in(self, params, duration, domain):
        domain_orig = domain.copy()
        for val in domain_orig: # val is week_hour_number
            # task must be within range
            if val in domain and val+duration not in params:
                domain.remove(val)
        return domain

    # task must start at day/time
    def apply_at(self, param_type, param,domain):
        domain_orig = domain.copy()
        for val in domain_orig:
            h, d = Day_Time().hour_day_split(val)
            if param_type == 'hour_number' and param != h:
                if val in domain: domain.remove(val)
            if param_type == 'day_number' and param != d:
                if val in domain: domain.remove(val)
            if param_type == 'week_hour_number' and param != val:
                if val in domain: domain.remove(val)
        return domain

    # soft deadline constraints: return cost to break constraint
    # ends-by implementation: domain_dt is the day, hour from the domain
    # constr_dt is the soft const spec, dur is the duration of task
    # soft_cost is the unit cost of completion delay
    # so if the tasks starts on domain_dt, it ends on domain_dt+dur
    """
    <t> ends-by <day> <time>, both must be specified
    delay = day_hour(T2) - day_hour(T1) + 24*(D2 - D1),
    where day_hour(9am) = 0, day_hour(5pm) = 7
    """
    def ends_by(self, domain_dt, constr_dt_str, dur, soft_cost):
        param_type,params = Day_Time().string_to_number_set(constr_dt_str)
        param_val = params.pop()
        dom_h, dom_d = Day_Time().hour_day_split(domain_dt+dur)
        if param_type == 'week_hour_number':
            con_h, con_d = Day_Time().hour_day_split(param_val)
            return 0 if domain_dt + dur <= param_val else soft_cost*(dom_h - con
        else:
```

```
        return None # not good, must be day and time

    def no_cost(self, day ,hour):
        return 0

    # hard binary constraint, the rest are implemented as gt, lt, eq
    def same_day(self, week_hour1, week_hour2):
        h1, d1 = Day_Time().hour_day_split(week_hour1)
        h2, d2 = Day_Time().hour_day_split(week_hour2)
        return d1 == d2

    # domain is a list of values
    def apply_hard_constraint(self, domain, duration, spec):
        tokens = func_key = spec.split(' ')
        if len(tokens) > 1:
            func_key = spec.split(' ')[0].strip()
            param_type, params = Day_Time().string_to_number_set(spec[len(func_ke
            if func_key == 'starts-before':
                # duration is 0 for starts before, since we do not modify the time
                return self.apply_before(param_type, params, 0, domain)
            if func_key == 'ends-before':
                return self.apply_before(param_type, params, duration, domain)
            if func_key == 'starts-after':
                return self.apply_after(param_type,params,0,domain)
            if func_key == 'ends-after':
                return self.apply_after(param_type, params, duration, domain)
            if func_key == 'starts-in':
                return self.apply_in(params, 0, domain)
            if func_key == 'ends-in':
                return self.apply_in(params, duration, domain)
        else:
            # here we have task day or time, it has no func key so we need to par
            param_type,params = Day_Time().string_to_week_hour_number(spec)
            return self.apply_at(param_type, params, domain)

    def get_cost_function(self, spec):
        func_dict = {'ends-by':self.ends_by, 'no-cost':self.no_cost}
        return [func_dict[spec]]

    # spec is the text of a constraint, e.g. 't1 before t2'
    # durations are durations of all tasks
    def get_binary_constraint(self, spec, durations):
        tokens = spec.strip().split(' ')
        if len(tokens) != 3: return None # error in spec
        # task1 relation task2
        fun_dict = {'before':le, 'after':ge, 'starts-at':eq, 'same-day':self.sam
        return Task_Constraint((tokens[0].strip(), tokens[2].strip()), spec, fun

    def get_CSP_with_Cost(self, input_lines):
        # Note: It would be more elegant to make task a class but AIpython is no
        # CSP_with_Cost inherits from CSP, which takes domains and constraints f
        domains = dict()
        constraints = []
        cost_functions = dict()
        durations = dict() # durations of tasks
        soft_day_time = dict() # day time specs of soft constraints
        soft_costs = dict() # costs of soft constraints

        for input_line in input_lines:
            func_spec = None
```

```
            input_line_tokens = input_line.strip().split(',')
            if len(input_line_tokens) != 2:
                return None # must have number of tokens = 2
            line_token1 = input_line_tokens[0].strip()
            line_token2 = input_line_tokens[1].strip()
            if line_token1 == 'task':
                tokens = line_token2.split(' ')
                if len(tokens) != 2:
                    return None # must have number of tokens = 3
                key = tokens[0].strip()
                # check the duration and save it
                duration = int(tokens[1].strip())
                if duration > Day_Time().num_hours_in_day:
                    return None
                durations[key] = duration
                # set zero cost function for this task as default, may add real
                cost_functions[key] = self.get_cost_function('no-cost')
                soft_costs[key] = '0'
                soft_day_time[key] = 'fri 5pm'
                # restrict domain to times that are within allowed range
                # that is start 9-5, start+duration in 9-5
                domains[key] = {x for x in self.default_domain \
                                if Day_Time().day_number(x+duration) \
                                == Day_Time().day_number(x)}
            elif line_token1 == 'domain':
                tokens = line_token2.split(' ')
                if len(tokens) < 2:
                    return None # must have number of tokens >= 2
                key = tokens[0].strip()
                # if soft constraint, it is handled differently from hard constr
                if tokens[1].strip() == 'ends-by':
                    # need to retain day time and cost from the line
                    # must have task, 'end-by', day, time, cost
                    # or task, 'end-by', day, cost
                    # or task, 'end-by', time, cost
                    if len(tokens) != 5:
                        return None
                    # get the rest of the line after 'ends-by'
                    soft_costs[key] = int(tokens[len(tokens)-1].strip()) # last
                    # pass the day time string to avoid passing param_type
                    day_time_str = tokens[2] + ' ' + tokens[3]
                    soft_day_time[key] = day_time_str
                    cost_functions[key] = self.get_cost_function(tokens[1].strip
                else:
                    # the rest of domain spec, after key, are hard unary domain
                    # func spec has day time, we also need duration
                    dur = durations[key]
                    func_spec = line_token2[len(key):].strip()
                    domains[key] = self.apply_hard_constraint(domains[key], dur,
            elif line_token1 == 'constraint': # all binary constraints
                constraints.append(self.get_binary_constraint(line_token2, durat
            else:
                return None

        return CSP_with_Cost(domains, durations, constraints, cost_functions, so

def create_CSP_from_spec(spec: str):
    input_lines = list()
    spec = spec.split('\n')
    # strip comments
```

```
    for input_line in spec:
        input_line = input_line.split('#')
        if len(input_line[0]) > 0:
            input_lines.append(input_line[0])
            print(input_line[0])
    # construct initial CSP problem
    csp = CSP_builder()
    csp_problem = csp.get_CSP_with_Cost(input_lines)
    return csp_problem
```

5. Greedy Search Constraint Solver using Domain Splitting and Arc Consistency

Create a GreedySearcher to search over the CSP.

The *cost* function for CSP nodes is used as the heuristic, but is actually a direct estimate of the total path cost function *f* used in A* Search.

In [7]:
```
from searchGeneric import AStarSearcher

class GreedySearcher(AStarSearcher):
    """ returns a searcher for a problem.
    Paths can be found by repeatedly calling search().
    """
    def add_to_frontier(self, path):
        """ add path to the frontier with the appropriate cost """
        # value = path.cost + self.problem.heuristic(path.end()) -- A* definitio
        value = path.end().cost
        self.frontier.add(path, value)
```

Run the GreedySearcher on the CSP derived from the sample input.

Note: The solution cost will always be 0 (which is wrong for the sample input) until you write the cost function in the cell above.

In [8]:
```
# Sample problem specification

sample_spec = """
# two tasks with two binary constraints and soft deadlines
task, t1 3
task, t2 4
# two binary constraints
constraint, t1 before t2
constraint, t1 same-day t2
# domain constraint
domain, t2 mon
# soft deadlines
domain, t1 ends-by mon 3pm 10
domain, t2 ends-by mon 3pm 10
"""
```

In [9]:
```
# display details (0 turns off)
Con_solver.max_display_level = 0
Search_with_AC_from_Cost_CSP.max_display_level = 2
GreedySearcher.max_display_level = 0
```

```
def test_csp_solver(searcher):
    final_path = searcher.search()
    if final_path == None:
        print('No solution')
    else:
        domains = final_path.end().domains
        result_str = ''
        for name, domain in domains.items():
            for n in domain:
                result_str += '\n'+str(name)+': '+Day_Time().week_hour_number_to
        print(result_str[1:]+'\ncost: '+str(final_path.end().cost))

csp_problem = create_CSP_from_spec(sample_spec)
solver = GreedySearcher(Search_with_AC_from_Cost_CSP(csp_problem))
test_csp_solver(solver)
```

```
task, t1 3
task, t2 4
constraint, t1 before t2
constraint, t1 same-day t2
domain, t2 mon
domain, t1 ends-by mon 3pm 10
domain, t2 ends-by mon 3pm 10
t1: mon 9am
t2: mon 12pm
cost: 10
```

6. Depth-First Search Constraint Solver

The Depth-First Constraint Solver in AIPython by default uses a random ordering of the variables in the CSP.

We need to modify this code to make it compatible with the arc consistency solver.

Run the solver by calling `dfs_solve1` (first solution) or `dfs_solve_all` (all solutions).

In [10]:
```
num_expanded = 0
display = False

def dfs_solver(constraints, domains, context, var_order):
    """ generator for all solutions to csp
        context is an assignment of values to some of the variables
        var_order is a list of the variables in csp that are not in context
    """
    global num_expanded, display
    to_eval = {c for c in constraints if c.can_evaluate(context)}
    if all(c.holds(context) for c in to_eval):
        if var_order == []:
            print("Nodes expanded to reach solution:", num_expanded)
            yield context
        else:
            rem_cons = [c for c in constraints if c not in to_eval]
            var = var_order[0]
            for val in domains[var]:
                if display:
                    print("Setting", var, "to", val)
                num_expanded += 1
                yield from dfs_solver(rem_cons, domains, context|{var:val}, var_
```

```
def dfs_solve_all(csp, var_order=None):
    """ depth-first CSP solver to return a list of all solutions to csp """
    global num_expanded
    num_expanded = 0
    if var_order == None:    # use an arbitrary variable order
        var_order = list(csp.domains)
    return list(dfs_solver(csp.constraints, csp.domains, {}, var_order))

def dfs_solve1(csp, var_order=None):
    """ depth-first CSP solver """
    global num_expanded
    num_expanded = 0
    if var_order == None:    # use an arbitrary variable order
        var_order = list(csp.domains)
    for sol in dfs_solver(csp.constraints, csp.domains, {}, var_order):
        return sol  # return first one
```

Run the Depth-First Solver on the sample problem.

Note: Again there are no costs calculated.

In [11]:
```
def test_dfs_solver(csp_problem):
    solution = dfs_solve1(csp_problem)
    if solution == None:
        print('No solution')
    else:
        result_str = ''
        for name in solution.keys():
            result_str += '\n'+str(name)+': '+Day_Time().week_hour_number_to_day
        print(result_str[1:])

# call the Depth-First Search solver
csp_problem = create_CSP_from_spec(sample_spec)
test_dfs_solver(csp_problem) # set display to True to see nodes expanded
```

```
task, t1 3
task, t2 4
constraint, t1 before t2
constraint, t1 same-day t2
domain, t2 mon
domain, t1 ends-by mon 3pm 10
domain, t2 ends-by mon 3pm 10
Nodes expanded to reach solution: 5
t1: mon 9am
t2: mon 12pm
```

7. Depth-First Search Constraint Solver using Forward Checking with MRV Heuristic

The Depth-First Constraint Solver in AIPython by default uses a random ordering of the variables in the CSP.

We redefine the `dfs_solver` methods to implement the MRV (Minimum Remaining Values) heuristic using forward checking.

Because the AIPython code is designed to manipulate domain sets, we also need to redefine `can_evaluate` to handle partial assignments.

In [12]:

```
num_expanded = 0
display = False

def can_evaluate(c, assignment):
    """ assignment is a variable:value dictionary
        returns True if the constraint can be evaluated given assignment
    """
    return assignment != {} and all(v in assignment.keys() and type(assignment[v

def mrv_dfs_solver(constraints, domains, context, var_order):
    """ generator for all solutions to csp.
        context is an assignment of values to some of the variables.
        var_order is a list of the variables in csp that are not in context.
    """
    global num_expanded, display
    if display:
        print("Context", context)
    to_eval = {c for c in constraints if can_evaluate(c, context)}
    if all(c.holds(context) for c in to_eval):
        if var_order == []:
            print("Nodes expanded to reach solution:", num_expanded)
            yield context
        else:
            rem_cons = [c for c in constraints if c not in to_eval] # constraint
            var = var_order[0]
            rem_vars = var_order[1:]
            for val in domains[var]:
                if display:
                    print("Setting", var, "to", val)
                num_expanded += 1
                rem_context = context|{var:val}
                # apply forward checking on remaining variables
                if len(var_order) > 1:
                    rem_vars_original = list((v, list(domains[v].copy())) for v
                    if display:
                        print("Original domains:", rem_vars_original)
                    # constraints that can't already be evaluated in rem_cons
                    rem_cons_ff = [c for c in constraints if c in rem_cons and n
                    for rem_var in rem_vars:
                        # constraints that can be evaluated by adding a value of
                        any_value = list(domains[rem_var])[0]
                        rem_to_eval = {c for c in rem_cons_ff if can_evaluate(c,
                        # new domain for rem_var are the values for which all ne
                        rem_vals = domains[rem_var].copy()
                        for rem_val in domains[rem_var]:
                            # no constraint with rem_var in the existing context
                            for c in rem_to_eval:
                                if not c.holds(rem_context|{rem_var: rem_val}):
                                    if rem_val in rem_vals:
                                        rem_vals.remove(rem_val)
                        domains[rem_var] = rem_vals
                        # order remaining variables by MRV
                        rem_vars.sort(key=lambda v: len(domains[v]))
                    if display:
                        print("After forward checking:", list((v, domains[v]) fo
                if rem_vars == [] or all(len(domains[rem_var]) > 0 for rem_var i
                    yield from mrv_dfs_solver(rem_cons, domains, context|{var:va
                # restore original domains if changed through forward checking
                if len(var_order) > 1:
```

```
                    if display:
                        print("Restoring original domain", rem_vars_original)
                    for (v, domain) in rem_vars_original:
                        domains[v] = domain
            if display:
                print("Nodes expanded so far:", num_expanded)

def mrv_dfs_solve_all(csp, var_order=None):
    """ depth-first CSP solver to return a list of all solutions to csp """
    global num_expanded
    num_expanded = 0
    if var_order == None:    # order variables by MRV
        var_order = list(csp.domains)
        var_order.sort(key=lambda var: len(csp.domains[var]))
    return list(mrv_dfs_solver(csp.constraints, csp.domains, {}, var_order))

def mrv_dfs_solve1(csp, var_order=None):
    """ depth-first CSP solver """
    global num_expanded
    num_expanded = 0
    if var_order == None:    # order variables by MRV
        var_order = list(csp.domains)
        var_order.sort(key=lambda var: len(csp.domains[var]))
    for sol in mrv_dfs_solver(csp.constraints, csp.domains, {}, var_order):
        return sol  # return first one
```

Run this solver on the sample problem.

Note: Again there are no costs calculated.

In [13]:
```
def test_mrv_dfs_solver(csp_problem):
    solution = mrv_dfs_solve1(csp_problem)
    if solution == None:
        print('No solution')
    else:
        result_str = ''
        for name in solution.keys():
            result_str += '\n'+str(name)+': '+Day_Time().week_hour_number_to_day
        print(result_str[1:])

# call the Depth-First MRV Search solver
csp_problem = create_CSP_from_spec(sample_spec)
test_mrv_dfs_solver(csp_problem) # set display to True to see nodes expanded
```

```
task, t1 3
task, t2 4
constraint, t1 before t2
constraint, t1 same-day t2
domain, t2 mon
domain, t1 ends-by mon 3pm 10
domain, t2 ends-by mon 3pm 10
Nodes expanded to reach solution: 5
t2: mon 12pm
t1: mon 9am
```

Assignment

Name: YINGLIN CHEN

zID: z5506175

Question 1 (4 marks)

Consider the search spaces for the fuzzy scheduling CSP solvers – domain splitting with arc consistency and the DFS solver (without forward checking).

- Describe the search spaces in terms of start state, successor functions and goal state(s) (1 mark)
- What is the branching factor and maximum depth to find any solution for the two algorithms (ignoring costs)? (1 mark)
- What is the worst case time and space complexity of the two search algorithms? (1 mark)
- Give one example of a fuzzy scheduling problem that is *easier* for the domain splitting with arc consistency solver than it is for the DFS solver, and explain why (1 mark)

For the second and third part-questions, give the answer in a general form in terms of fuzzy scheduling CSP size parameters.

Answers for Question 1

- The search space consists of starting from the start state, generating successor states by assignment, and finally reaching the goal state with all variable assignments and satisfying constraints. The successor functions assigns values to the variables to generate new nodes, and the goal state is the set of all the conditions that are satisfied.
- For DFS, each decision assigns one variable a value from its domain, so the branching factor equals the domain size d, and maxium depth equals the number of variables n. For domain splitting with arc consistency, instead of directly assigning values, the value domain is gradually divided into subdomains, and only about 2 branches are generated each time, so the branching factor is usually 2. The maxiumu depth is O(nlogd).'d' is size of domain, 'n' is the number of constraints.
- For DFS, time complexity is O(d^n),space complexity is O(n). 'd' is size of domain, 'n' is the number of constraints. For domain splitting with arc consistency, time complexity is O(cd^3), space complexity is O(cd). 'c' is the constrain the number of arcs, 'd' is domain size per variable.
- For example, have four tasks task(t1,t2,t3,t4),and have three binary constraints (constraint, t4 starts-at t3), (constraint, t1 after t3),(constraint, t2 after t1). With these three binary constraints, domain splitting with arc consistency solver can quickly find the range of each task and remove invalid time, for example, it is easy to know that t2 is ranked last. With DFS, however, it would have to try all t2 options and then backtrack.

Question 2 (5 marks)

Define the *cost* function for a fuzzy scheduling CSP (i.e. a node in the search space for domain splitting and arc consistency) as the total cost of the soft deadline constraints violated for all of the variables, assuming that each variable is assigned one of the best possible values from its domain, where a "best" value for a variable *v* is one that has the lowest cost to violate the soft deadline constraint (if any) for that variable *v*.

- Implement the cost function in the indicated cell and place a copy of the code below (3 marks)
- What is its computational complexity (give a general form in terms of fuzzy scheduling CSP size parameters)? (1 mark)
- Show that the cost function *f* never decreases along a path, and explain why this means the search algorithm is optimal (1 mark)

In [ ]:
```
# Code for Question 2
# Place a copy of your code here and run it in the relevant cell
def calculate_cost(self):
        """ this is really a function f = path cost + heuristic to be used by th
        cost = 0
        for var in self.variables:
            dom = self.domains[var]
            is_cost = self.cost_functions[var][0]
            if is_cost.__name__ == 'no_cost':
                continue
            duration = self.durations[var]
            soft_time = self.soft_day_time[var]
            soft_cost = self.soft_costs[var]
            min_cost = None
            for i in dom:
                penalty = is_cost(i, soft_time, duration, soft_cost)
                if min_cost is None or penalty < min_cost:
                    min_cost = penalty
            if min_cost is None:
                min_cost = 0
            cost += min_cost
        return cost
```

Answers for Question 2

- time complexity is O(dn),space complexity is O(1). 'd' is size of domain, 'n' is the number of constraints.
- In fuzzy scheduling CSP, each step of search will make the optional time of each variable less, and for each task, only the lowest soft constraint cost is found in the current optional time. When the search continues, the reduction of the field will only make the minimum cost unchanged or become higher, but there is no reduction. So the cost function *f* never decreases along a path and the optimal solution is found the first time.

Question 3 (4 marks)

Conduct an empirical evaluation of the domain splitting CSP solver using the cost function defined as above compared to using no cost function (i.e. the zero cost function,

as originally defined in the above cell). Use the *average number of nodes expanded* as a metric to compare the two algorithms.

- Write a function `generate_problem(n)` that takes an integer `n` and generates a problem specification with `n` tasks and a random set of hard constraints and soft deadline constraints in the correct format for the constraint solvers (2 marks)

Run the CSP solver (with and without the cost function) over a number of problems of size `n` for a range of values of `n`.

- Plot the performance of the two constraint solving algorithms on the above metric against `n` (1 mark)
- Quantify the performance gain (if any) achieved by the use of this cost function (1 mark)

In [136…

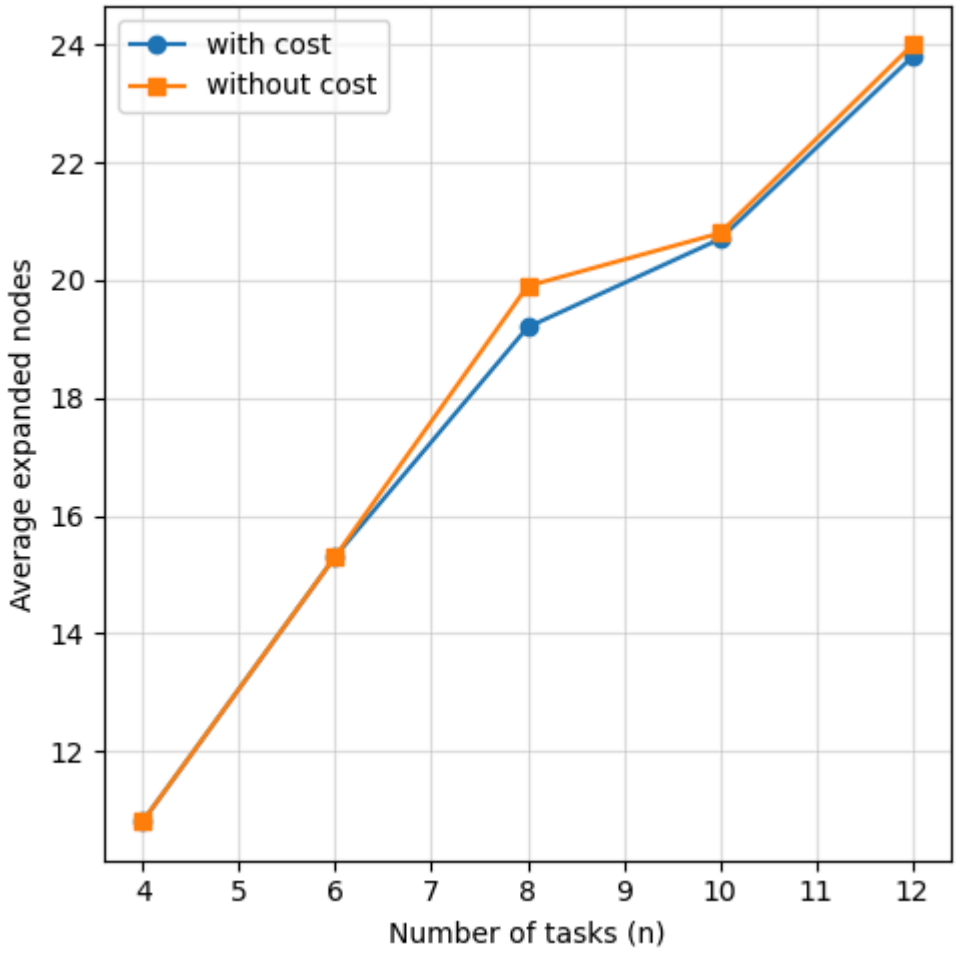
```python
# Code for Question 3
# Place your code here
import random

def generate_problem(n, seed = None):
    if seed is None:
        random.seed(seed)
    spec_info = []
    # --- random tasks ---
    for i in range(1, n + 1):
        duration = random.randint(1, 4)
        spec_info.append(f"task, t{i} {duration}")
    # --- random constraint's number ----
    constraint_types = ['before', 'after', 'same-day', 'starts-at']
    tasks = [f"t{i}" for i in range(1, n + 1)]
    constraints_num = random.randint(max(0, n // 3), max(1, (2 * n) // 3))
    added_constraints = set()
    for _ in range(constraints_num):
        t1, t2 = random.sample(tasks, 2)
        rel = random.choice(constraint_types)
        if rel == 'same-day':
            key = tuple(sorted([rel, t1, t2]))
        else:
            key = (rel, t1, t2)
        if key in added_constraints:
            continue
        added_constraints.add(key)
        spec_info.append(f"constraint, {t1} {rel} {t2}")
    # --- random hard constraints'number -----
    days  = ['mon','tue','wed','thu','fri']
    times = ['9am','10am','11am','12pm','1pm','2pm','3pm','4pm']
    hc_types = ['starts-before', 'starts-after', 'ends-before', 'ends-after']
    hard_constraints_num = random.randint(0, n // 4)
    for _ in range(hard_constraints_num):
        task_id = random.choice(tasks)
        hc_type  = random.choice(hc_types)
        day  = random.choice(days)
        time = random.choice(times)
        spec_info.append(f"domain, {task_id} {hc_type} {day} {time}")
    # --- random soft constraints' number
    soft_constraints_num = random.randint(n // 2, max(n // 2 + 1, int(n * 0.8)))
    tasks_with_soft = random.sample(tasks, min(soft_constraints_num, len(tasks))
```

```
    for task_id in tasks_with_soft:
        day  = random.choice(days)
        time = random.choice(times)
        cost = random.randint(5,20)
        spec_info.append(f"domain, {task_id} ends-by {day} {time} {cost}")

    return '\n'.join(spec_info)

class Greedy_Search_Counted(Search_with_AC_from_Cost_CSP):
    def __init__(self, csp):
        super().__init__(csp)
        self.nodes_added = 0

    def neighbors(self, node):
        self.nodes_added += 1
        return super().neighbors(node)

Con_solver.max_display_level = 0
Search_with_AC_from_Cost_CSP.max_display_level = 0

def solve_with_cost(spec):
    csp_problem = create_CSP_from_spec(spec)
    solver = GreedySearcher(Greedy_Search_Counted(csp_problem))
    solver.search()
    expanded_nodes = solver.problem.nodes_added
    return expanded_nodes

def solve_without_cost(spec):
    csp_problem = create_CSP_from_spec(spec)
    for i in csp_problem.cost_functions:
        csp_problem.cost_functions[i] = [CSP_builder.no_cost]

    solver = GreedySearcher(Greedy_Search_Counted(csp_problem))
    solver.search()
    expanded_nodes = solver.problem.nodes_added
    return expanded_nodes
```

Answers for Question 3

- According to the obtained results, it can be seen that the search efficiency of CSP is only slightly improved after adding the cost function, and this improvement is also found to be very unstable through the experimental process. It shows that although the cost function has some guiding effect, the efficiency of the search is mainly determined by the hard constraints and ARC in the fuzzy scheduling problem.

task_num	with_cost	without_cost	gain %
4	8.3	8.6	3.5%
6	8.4	8.6	2.3%
8	18.7	19.1	2.1%
10	26.9	27.1	0.7%
12	33.4	33.9	1.5%



Question 4 (5 marks)

Compare the Depth-First Search (DFS) solver to the Depth-First Search solver using forward checking with Minimum Remaining Values heuristic (DFS-MRV). For this question, ignore the costs associated with the CSP problems.

- What is the worst case time and space complexity of each algorithm (give a general form in terms of fuzzy scheduling problem sizes)? (1 mark)
- What are the properties of the search algorithms (completeness, optimality)? (1 mark)
- Give an example of a problem that is *easier* for the DFS-MRV solver than it is for the DFS solver, and explain why (1 mark)
- Empirically compare the quality of the first solution found by DFS and DFS-MRV compared to the optimal solution (1 mark)

- Empirically compare DFS-MRV with DFS in terms of the number of nodes expanded (1 mark)

For the empirical evaluations, run the two algorithms on a variety of problems of size `n` for varying `n`. Note that the domain splitting CSP solver with costs should always find an optimal solution.

In [132…

```
# Code for Question 4
# Place your code here
import random

def assignment_cost(csp, assignment):
    total = 0
    for v, val in assignment.items():
        is_cost = csp.cost_functions[v][0]
        if is_cost.__name__ == 'no_cost':
            continue
        total += is_cost(val, csp.soft_day_time[v], csp.durations[v], csp.soft_c
    return total

def dfs_vs_mrv(spec):
    dfs_csp = create_CSP_from_spec(spec)
    mrv_csp = create_CSP_from_spec(spec)
    s1 = dfs_solve1(dfs_csp)
    dfs_nodes = num_expanded
    dfs_first_cost = assignment_cost(dfs_csp, s1) if s1 else float('inf')
    s2 = mrv_dfs_solve1(mrv_csp)
    mrv_nodes = num_expanded
    mrv_first_cost = assignment_cost(mrv_csp, s2) if s2 else float('inf')
    return dfs_nodes, mrv_nodes, dfs_first_cost, mrv_first_cost
```

Answers for Question 4

- For DFS, time complexity is O(d^n),space complexity is O(n). For DFS-MRV, time complexity is O(d^n),space complexity is O(nd).'d' is size of domain, 'n' is the number of constraints.
- Both a and b are completeness, in the sense that both algorithms can find a solution given a finite search space. But both of them do not have optimality, they just find a feasible solution, and do not consider the minimum cost problem, although this problem says to ignore the cost, but I think they do not have optimality.
- DFS-MRV is better when there are more strong constraints between tasks. For example, in the map coloring problem, DFS-MRV will select the neighboring variable with small domain first, because the few options can expose constraint conflicts at an earlier stage and thus prune ahead of time. While DFS may choose irrelevant variables first, such as selecting a coloring that is not adjacent to the current one on the map coloring, if this branch conflicts, dfs has to go a lot of dead ends before it finds out.
- According to the content of the picture, it can be known that DFS-MRV does not reduce the quality of the solution. In this experiment, the cost of the first solution found by DFS and DFS-MRV is exactly the same. This indicates that although MRV changes the search order, it does not sacrifice the feasibility or rationality of the solution.

- According to the content of the picture, with the increase of the task number 'n', the number of nodes in DFS grows explosively while DFS-MRV always maintains a very small search order. DFS-MRV is very effective in such large-scale solutions, especially when there are strong constraints or large-scale variables. It finds conflicts in advance through the minimum residual value strategy and avoids a large number of invalid branches.

n	DFS avg	DFS-MRV avg	DFS_cost	MRV_cost	speedup
2	14.6	2.6	7.2	7.2	5.62
4	28.8	5.6	25.6	25.6	5.14
6	12268.2	6.8	0.0	0.0	1804.15

Question 5 (4 marks)

The DFS solver chooses variables in random order, and systematically explores all values for those variables in no particular order.

Incorporate costs into the DFS constraint solver as heuristics to guide the search. Similar to the cost function for the domain splitting solver, for a given variable *v*, the cost of assigning the value *val* to *v* is the cost of violating the soft deadline constraint (if any) associated with *v* for the value *val*. The *minimum cost* for *v* is the lowest cost from amongst the values in the domain of *v*. The DFS solver should choose a variable *v* with lowest minimum cost, and explore its values in order of cost from lowest to highest.

- Implement this behaviour by modifying the code in `dfs_solver` and place a copy of the code below (2 marks)
- Empirically compare the performance of DFS with and without these heuristics (2 marks)

For the empirical evaluations, again run the two algorithms on a variety of problems of size `n` for varying `n`.

In [115…

```
# Code for Question 5
# Place a copy of your code here and run it in the relevant cell
def cost_value(var, val, cost_functions, soft_day_time, durations, soft_costs):
    func = None
    if var in cost_functions and cost_functions[var]:
        func = cost_functions[var][0]
    if func.__name__ == 'no_cost':
        return 0
    return func(val, soft_day_time[var], durations[var], soft_costs[var])

def min_cost(var, domains, cost_functions, soft_day_time, durations, soft_costs)
    if not domains[var]:
        return 0
    min_cost = None
    for a in domains[var]:
        c = cost_value(var, a, cost_functions, soft_day_time, durations, soft_co
        if min_cost is None or c < min_cost:
            min_cost = c
    return min_cost
```

```
def cost_dfs_solver(constraints, domains, context, var_order,
                            cost_functions, soft_day_time, soft_costs, durations)
    global num_expanded, display
    to_eval = {c for c in constraints if c.can_evaluate(context)}
    if all(c.holds(context) for c in to_eval):
        if var_order == []:
            print("Nodes expanded to reach solution:", num_expanded)
            yield context
        else:
            rem_cons = [c for c in constraints if c not in to_eval]
            var = min(var_order,key=lambda v: (min_cost(v, domains, cost_functio
            next_vars = [v for v in var_order if v != var]
            ordered_vals = sorted(domains[var],key=lambda a: (cost_value(var, a,
            for val in ordered_vals:
                if display:
                    print("Setting", var, "to", val)
                num_expanded += 1
                new_ctx = context | {var: val}
                yield from cost_dfs_solver(
                    rem_cons, domains, new_ctx, next_vars,
                    cost_functions, soft_day_time, soft_costs, durations
                )

def cost_dfs_solve1(csp, var_order=None):
    global num_expanded
    num_expanded = 0
    if var_order is None:
        var_order = list(csp.domains)
    for sol in cost_dfs_solver(
        csp.constraints, csp.domains, {}, var_order,
        csp.cost_functions, csp.soft_day_time, csp.soft_costs, csp.durations
    ):
        return sol
```

Answers for Question 5

- After adding the soft constraint cost heuristic to DFS, the variable and its value with the lowest cost are selected first. Compared with the original DFS, the heuristic DFS significantly reduces the number of node expansions in most cases, especially when n is small or the cost difference is obvious. However, the effect is very unstable. When the constraint difference is small, the efficiency of both DFS and heuristic DFS is similar, and even worse in some cases.

n	DFS avg	DFS+cost avg	gain %
4	4.0	4.0	0.0%
6	9817.0	6500.0	33.8%
8	9946.0	1617.0	83.7%

Question 6 (3 marks)

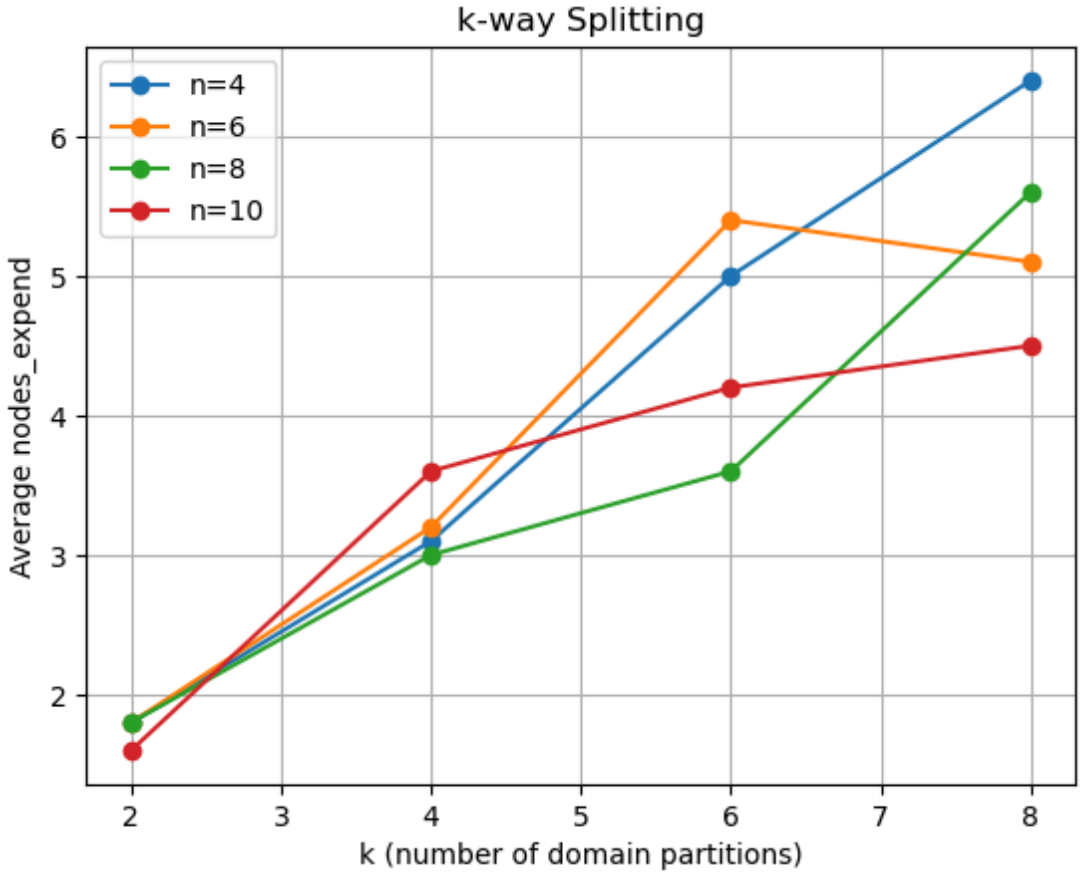
The CSP solver with domain splitting splits a CSP variable domain into *exactly two* partitions. Poole & Mackworth claim that in practice, this is as good as splitting into a larger number of partitions. In this question, empirically evaluate this claim for fuzzy scheduling CSPs.

- Write a new `partition_domain` function that partitions a domain into a list of `k` partitions, where `k` is a parameter to the function (1 mark)
- Modify the CSP solver to use the list of `k` partitions and evaluate the performance of the solver using the above metric for a range of values of `k` (2 marks)

In [63]:
```
# Code for Question 6
# Place a copy of your code here and run it in the relevant cell
def partition_domain_k(dom, k):
    dom_list = list(dom)
    n = len(dom_list)
    if n == 0:
        return []
    if k >= n:
        return [[val] for val in dom_list]
    base_size = n // k
    remainder = n % k
    partitions = []
    index = 0
    for i in range(k):
        size = base_size + (1 if i < remainder else 0)
        sub_domain = dom_list[index : index + size]
        partitions.append(sub_domain)
        index += size
    return partitions

class Search_with_AC_from_Cost_CSP_Ksplit(Search_with_AC_from_Cost_CSP):
    def __init__(self, csp, k=2):
        super().__init__(csp)
        self.k = k
        self.nodes_expend = 0
    def neighbors(self, node):
        neighs = []
        var = select(x for x in node.domains if len(node.domains[x]) > 1)
        if var:
            parts = partition_domain_k(node.domains[var], self.k)
            self.display(2, f"Splitting {var} into {len(parts)} parts: {parts}")
            to_do = self.cons.new_to_do(var, None)
            for dom in parts:
                newdoms = node.domains | {var: dom}
                cons_doms = self.cons.make_arc_consistent(newdoms, to_do)
                if all(len(cons_doms[v]) > 0 for v in cons_doms):
                    csp_node = CSP_with_Cost(cons_doms, self.durations, self.con
                                self.cost_functions, self.soft_day_time, self.soft_
                    neighs.append(Arc(node, csp_node))
                    self.nodes_expend += 1
                else:
                    self.display(2,"...",var,"in",dom,"has no solution")
        return neighs
```

Answers for Question 6

From this figure, we can see that although splitting the domain into more regions provides more precise control, in practice it does not improve performance and may complicate the problem by increasing the branching factor. Two partitions is sufficient because arc consistency already eliminates a lot of inconsistencies, so adding more partitions won't improve the performance significantly.



In [135…

```python
# Q3 extra code
import matplotlib.pyplot as plt

def statistical_data_q3():
    ns = [4, 6, 8, 10, 12]
    trials = 10
    rnd = random.Random(0)
    avg_with_cost, avg_without_cost = [], []

    for n in ns:
        sum_with_cost = 0
        sum_without_cost = 0
        for _ in range(trials):
            spec = generate_problem(n, seed=rnd.randint(0, 1000))
            sum_with_cost += solve_with_cost(spec)
            sum_without_cost += solve_without_cost(spec)

        avg_with_cost.append(sum_with_cost / trials)
        avg_without_cost.append(sum_without_cost / trials)

    return ns, avg_with_cost, avg_without_cost

def output_result(ns, avg_with, avg_without):
    print(f'{"task_num":>10} | {"with_cost":>12} | {"without_cost":>12} | {"gain
```

```
    print('-' * 44)
    for n, w, z in zip(ns, avg_with, avg_without):
        gain = 0.0 if z == 0 else (z - w) / z * 100.0
        print(f'{n:>10} | {w:12.1f} | {z:12.1f} | {gain:6.1f}%')

ns, avg_with, avg_without = statistical_data_q3()
output_result(ns, avg_with, avg_without)

def plot_results(ns, avg_with, avg_without):
    plt.figure(figsize=(5, 5))
    plt.plot(ns, avg_with,    marker='o', label='with cost')
    plt.plot(ns, avg_without, marker='s', label='without cost')
    plt.xlabel('Number of tasks (n)')
    plt.ylabel('Average expanded nodes')
    plt.grid(True, alpha=0.4)
    plt.legend()
    plt.tight_layout()
    plt.show()

plot_results(ns, avg_with, avg_without)
```

```
task, t1 2
task, t2 1
task, t3 2
task, t4 1
constraint, t4 same-day t2
domain, t4 ends-before wed 9am
domain, t1 ends-by thu 11am 11
domain, t2 ends-by mon 4pm 15
domain, t4 ends-by wed 12pm 7
task, t1 2
task, t2 1
task, t3 2
task, t4 1
constraint, t4 same-day t2
domain, t4 ends-before wed 9am
domain, t1 ends-by thu 11am 11
domain, t2 ends-by mon 4pm 15
domain, t4 ends-by wed 12pm 7
task, t1 4
task, t2 2
task, t3 2
task, t4 4
constraint, t4 before t2
constraint, t3 starts-at t2
domain, t2 ends-by fri 10am 12
domain, t4 ends-by tue 10am 19
domain, t1 ends-by fri 2pm 5
task, t1 4
task, t2 2
task, t3 2
task, t4 4
constraint, t4 before t2
constraint, t3 starts-at t2
domain, t2 ends-by fri 10am 12
domain, t4 ends-by tue 10am 19
domain, t1 ends-by fri 2pm 5
task, t1 4
task, t2 1
task, t3 2
task, t4 2
constraint, t3 before t1
constraint, t3 same-day t2
domain, t4 ends-by tue 1pm 14
domain, t1 ends-by thu 9am 14
task, t1 4
task, t2 1
task, t3 2
task, t4 2
constraint, t3 before t1
constraint, t3 same-day t2
domain, t4 ends-by tue 1pm 14
domain, t1 ends-by thu 9am 14
task, t1 3
task, t2 4
task, t3 4
task, t4 2
constraint, t4 after t2
domain, t4 starts-before wed 9am
domain, t1 ends-by tue 4pm 13
domain, t3 ends-by fri 11am 12
```

```
domain, t4 ends-by tue 9am 20
task, t1 3
task, t2 4
task, t3 4
task, t4 2
constraint, t4 after t2
domain, t4 starts-before wed 9am
domain, t1 ends-by tue 4pm 13
domain, t3 ends-by fri 11am 12
domain, t4 ends-by tue 9am 20
task, t1 3
task, t2 2
task, t3 3
task, t4 4
constraint, t2 before t1
domain, t1 starts-after thu 11am
domain, t2 ends-by mon 4pm 9
domain, t1 ends-by tue 4pm 16
domain, t4 ends-by fri 12pm 12
task, t1 3
task, t2 2
task, t3 3
task, t4 4
constraint, t2 before t1
domain, t1 starts-after thu 11am
domain, t2 ends-by mon 4pm 9
domain, t1 ends-by tue 4pm 16
domain, t4 ends-by fri 12pm 12
task, t1 2
task, t2 3
task, t3 4
task, t4 1
constraint, t3 after t2
constraint, t4 before t2
domain, t1 starts-before thu 12pm
domain, t2 ends-by wed 10am 15
domain, t4 ends-by thu 9am 10
task, t1 2
task, t2 3
task, t3 4
task, t4 1
constraint, t3 after t2
constraint, t4 before t2
domain, t1 starts-before thu 12pm
domain, t2 ends-by wed 10am 15
domain, t4 ends-by thu 9am 10
task, t1 2
task, t2 4
task, t3 2
task, t4 4
constraint, t1 same-day t4
constraint, t3 before t1
domain, t3 starts-after wed 1pm
domain, t1 ends-by thu 3pm 18
domain, t3 ends-by tue 9am 14
task, t1 2
task, t2 4
task, t3 2
task, t4 4
constraint, t1 same-day t4
```

```
constraint, t3 before t1
domain, t3 starts-after wed 1pm
domain, t1 ends-by thu 3pm 18
domain, t3 ends-by tue 9am 14
task, t1 2
task, t2 4
task, t3 2
task, t4 1
constraint, t2 same-day t3
domain, t1 starts-before thu 4pm
domain, t2 ends-by thu 11am 6
domain, t4 ends-by fri 3pm 9
task, t1 2
task, t2 4
task, t3 2
task, t4 1
constraint, t2 same-day t3
domain, t1 starts-before thu 4pm
domain, t2 ends-by thu 11am 6
domain, t4 ends-by fri 3pm 9
task, t1 1
task, t2 1
task, t3 1
task, t4 1
constraint, t2 same-day t4
domain, t1 ends-before mon 1pm
domain, t4 ends-by thu 3pm 9
domain, t3 ends-by fri 4pm 6
task, t1 1
task, t2 1
task, t3 1
task, t4 1
constraint, t2 same-day t4
domain, t1 ends-before mon 1pm
domain, t4 ends-by thu 3pm 9
domain, t3 ends-by fri 4pm 6
task, t1 3
task, t2 2
task, t3 4
task, t4 4
constraint, t2 before t3
domain, t4 ends-by tue 11am 15
domain, t2 ends-by thu 12pm 20
domain, t1 ends-by thu 1pm 10
task, t1 3
task, t2 2
task, t3 4
task, t4 4
constraint, t2 before t3
domain, t4 ends-by tue 11am 15
domain, t2 ends-by thu 12pm 20
domain, t1 ends-by thu 1pm 10
task, t1 4
task, t2 2
task, t3 1
task, t4 3
task, t5 2
task, t6 4
constraint, t4 starts-at t1
constraint, t1 before t5
```

```
constraint, t3 before t6
constraint, t6 same-day t2
domain, t3 ends-by thu 10am 8
domain, t5 ends-by thu 12pm 19
domain, t1 ends-by wed 11am 16
task, t1 4
task, t2 2
task, t3 1
task, t4 3
task, t5 2
task, t6 4
constraint, t4 starts-at t1
constraint, t1 before t5
constraint, t3 before t6
constraint, t6 same-day t2
domain, t3 ends-by thu 10am 8
domain, t5 ends-by thu 12pm 19
domain, t1 ends-by wed 11am 16
task, t1 4
task, t2 2
task, t3 3
task, t4 3
task, t5 4
task, t6 4
constraint, t6 after t2
constraint, t1 same-day t6
constraint, t3 starts-at t6
domain, t5 ends-by fri 2pm 20
domain, t2 ends-by tue 11am 7
domain, t6 ends-by thu 1pm 11
domain, t3 ends-by wed 1pm 12
task, t1 4
task, t2 2
task, t3 3
task, t4 3
task, t5 4
task, t6 4
constraint, t6 after t2
constraint, t1 same-day t6
constraint, t3 starts-at t6
domain, t5 ends-by fri 2pm 20
domain, t2 ends-by tue 11am 7
domain, t6 ends-by thu 1pm 11
domain, t3 ends-by wed 1pm 12
task, t1 3
task, t2 4
task, t3 4
task, t4 2
task, t5 3
task, t6 2
constraint, t6 same-day t4
constraint, t1 before t2
domain, t1 ends-by thu 10am 20
domain, t5 ends-by tue 9am 17
domain, t2 ends-by thu 10am 18
domain, t4 ends-by mon 11am 14
task, t1 3
task, t2 4
task, t3 4
task, t4 2
```

```
task, t5 3
task, t6 2
constraint, t6 same-day t4
constraint, t1 before t2
domain, t1 ends-by thu 10am 20
domain, t5 ends-by tue 9am 17
domain, t2 ends-by thu 10am 18
domain, t4 ends-by mon 11am 14
task, t1 4
task, t2 4
task, t3 2
task, t4 3
task, t5 2
task, t6 1
constraint, t3 same-day t4
constraint, t1 starts-at t5
constraint, t3 same-day t6
domain, t6 ends-by tue 4pm 16
domain, t1 ends-by thu 2pm 6
domain, t5 ends-by wed 4pm 10
domain, t4 ends-by mon 10am 15
task, t1 4
task, t2 4
task, t3 2
task, t4 3
task, t5 2
task, t6 1
constraint, t3 same-day t4
constraint, t1 starts-at t5
constraint, t3 same-day t6
domain, t6 ends-by tue 4pm 16
domain, t1 ends-by thu 2pm 6
domain, t5 ends-by wed 4pm 10
domain, t4 ends-by mon 10am 15
task, t1 4
task, t2 3
task, t3 2
task, t4 2
task, t5 3
task, t6 2
constraint, t5 before t6
constraint, t2 same-day t3
constraint, t2 before t6
domain, t6 starts-before tue 3pm
domain, t3 ends-by tue 9am 5
domain, t2 ends-by wed 10am 17
domain, t6 ends-by thu 11am 20
task, t1 4
task, t2 3
task, t3 2
task, t4 2
task, t5 3
task, t6 2
constraint, t5 before t6
constraint, t2 same-day t3
constraint, t2 before t6
domain, t6 starts-before tue 3pm
domain, t3 ends-by tue 9am 5
domain, t2 ends-by wed 10am 17
domain, t6 ends-by thu 11am 20
```

```
task, t1 1
task, t2 2
task, t3 4
task, t4 4
task, t5 4
task, t6 2
constraint, t1 before t5
constraint, t5 starts-at t2
constraint, t1 same-day t3
domain, t6 ends-before tue 12pm
domain, t4 ends-by tue 11am 7
domain, t5 ends-by mon 1pm 11
domain, t1 ends-by mon 4pm 20
task, t1 1
task, t2 2
task, t3 4
task, t4 4
task, t5 4
task, t6 2
constraint, t1 before t5
constraint, t5 starts-at t2
constraint, t1 same-day t3
domain, t6 ends-before tue 12pm
domain, t4 ends-by tue 11am 7
domain, t5 ends-by mon 1pm 11
domain, t1 ends-by mon 4pm 20
task, t1 3
task, t2 3
task, t3 3
task, t4 2
task, t5 4
task, t6 3
constraint, t1 starts-at t4
constraint, t5 starts-at t1
constraint, t4 after t1
domain, t1 ends-by wed 2pm 5
domain, t4 ends-by thu 4pm 8
domain, t6 ends-by tue 10am 5
No (more) solutions. Total of 1 paths expanded.
task, t1 3
task, t2 3
task, t3 3
task, t4 2
task, t5 4
task, t6 3
constraint, t1 starts-at t4
constraint, t5 starts-at t1
constraint, t4 after t1
domain, t1 ends-by wed 2pm 5
domain, t4 ends-by thu 4pm 8
domain, t6 ends-by tue 10am 5
No (more) solutions. Total of 1 paths expanded.
task, t1 1
task, t2 2
task, t3 1
task, t4 3
task, t5 1
task, t6 2
constraint, t3 after t4
constraint, t4 starts-at t1
```

```
domain, t5 ends-before wed 1pm
domain, t5 ends-by mon 4pm 6
domain, t2 ends-by thu 12pm 6
domain, t4 ends-by tue 3pm 6
task, t1 1
task, t2 2
task, t3 1
task, t4 3
task, t5 1
task, t6 2
constraint, t3 after t4
constraint, t4 starts-at t1
domain, t5 ends-before wed 1pm
domain, t5 ends-by mon 4pm 6
domain, t2 ends-by thu 12pm 6
domain, t4 ends-by tue 3pm 6
task, t1 3
task, t2 4
task, t3 3
task, t4 1
task, t5 3
task, t6 3
constraint, t5 before t3
constraint, t5 after t6
constraint, t5 same-day t3
domain, t3 ends-after fri 2pm
domain, t1 ends-by mon 3pm 15
domain, t4 ends-by wed 4pm 7
domain, t2 ends-by mon 10am 5
task, t1 3
task, t2 4
task, t3 3
task, t4 1
task, t5 3
task, t6 3
constraint, t5 before t3
constraint, t5 after t6
constraint, t5 same-day t3
domain, t3 ends-after fri 2pm
domain, t1 ends-by mon 3pm 15
domain, t4 ends-by wed 4pm 7
domain, t2 ends-by mon 10am 5
task, t1 1
task, t2 2
task, t3 4
task, t4 3
task, t5 2
task, t6 4
constraint, t1 starts-at t6
constraint, t1 same-day t2
domain, t2 ends-by tue 9am 14
domain, t6 ends-by mon 10am 8
domain, t4 ends-by tue 1pm 17
domain, t1 ends-by mon 4pm 14
task, t1 1
task, t2 2
task, t3 4
task, t4 3
task, t5 2
task, t6 4
```

```
constraint, t1 starts-at t6
constraint, t1 same-day t2
domain, t2 ends-by tue 9am 14
domain, t6 ends-by mon 10am 8
domain, t4 ends-by tue 1pm 17
domain, t1 ends-by mon 4pm 14
task, t1 4
task, t2 4
task, t3 2
task, t4 3
task, t5 1
task, t6 1
task, t7 4
task, t8 4
constraint, t2 after t8
constraint, t7 before t8
domain, t1 ends-after tue 2pm
domain, t4 ends-before thu 4pm
domain, t6 ends-by wed 3pm 12
domain, t4 ends-by thu 2pm 13
domain, t3 ends-by mon 2pm 20
domain, t5 ends-by mon 1pm 15
domain, t7 ends-by thu 11am 11
task, t1 4
task, t2 4
task, t3 2
task, t4 3
task, t5 1
task, t6 1
task, t7 4
task, t8 4
constraint, t2 after t8
constraint, t7 before t8
domain, t1 ends-after tue 2pm
domain, t4 ends-before thu 4pm
domain, t6 ends-by wed 3pm 12
domain, t4 ends-by thu 2pm 13
domain, t3 ends-by mon 2pm 20
domain, t5 ends-by mon 1pm 15
domain, t7 ends-by thu 11am 11
task, t1 3
task, t2 4
task, t3 4
task, t4 4
task, t5 4
task, t6 1
task, t7 1
task, t8 2
constraint, t1 starts-at t2
constraint, t8 starts-at t4
domain, t7 starts-before wed 9am
domain, t1 ends-by fri 4pm 15
domain, t2 ends-by tue 4pm 15
domain, t6 ends-by tue 1pm 7
domain, t3 ends-by wed 4pm 5
domain, t5 ends-by fri 12pm 15
task, t1 3
task, t2 4
task, t3 4
task, t4 4
```

```
task, t5 4
task, t6 1
task, t7 1
task, t8 2
constraint, t1 starts-at t2
constraint, t8 starts-at t4
domain, t7 starts-before wed 9am
domain, t1 ends-by fri 4pm 15
domain, t2 ends-by tue 4pm 15
domain, t6 ends-by tue 1pm 7
domain, t3 ends-by wed 4pm 5
domain, t5 ends-by fri 12pm 15
task, t1 2
task, t2 1
task, t3 3
task, t4 2
task, t5 2
task, t6 3
task, t7 4
task, t8 4
constraint, t1 same-day t6
constraint, t8 before t5
domain, t8 ends-after wed 4pm
domain, t2 ends-after fri 9am
domain, t4 ends-by wed 3pm 5
domain, t7 ends-by fri 1pm 15
domain, t8 ends-by wed 11am 16
domain, t6 ends-by tue 11am 15
domain, t3 ends-by fri 12pm 8
domain, t2 ends-by tue 2pm 15
task, t1 2
task, t2 1
task, t3 3
task, t4 2
task, t5 2
task, t6 3
task, t7 4
task, t8 4
constraint, t1 same-day t6
constraint, t8 before t5
domain, t8 ends-after wed 4pm
domain, t2 ends-after fri 9am
domain, t4 ends-by wed 3pm 5
domain, t7 ends-by fri 1pm 15
domain, t8 ends-by wed 11am 16
domain, t6 ends-by tue 11am 15
domain, t3 ends-by fri 12pm 8
domain, t2 ends-by tue 2pm 15
task, t1 4
task, t2 1
task, t3 1
task, t4 3
task, t5 1
task, t6 3
task, t7 3
task, t8 4
constraint, t6 before t7
constraint, t5 same-day t2
constraint, t7 after t3
constraint, t3 same-day t7
```

```
domain, t5 starts-before fri 3pm
domain, t1 ends-by fri 12pm 17
domain, t2 ends-by wed 11am 6
domain, t7 ends-by thu 9am 7
domain, t3 ends-by fri 12pm 14
domain, t5 ends-by tue 2pm 8
domain, t4 ends-by mon 9am 12
task, t1 4
task, t2 1
task, t3 1
task, t4 3
task, t5 1
task, t6 3
task, t7 3
task, t8 4
constraint, t6 before t7
constraint, t5 same-day t2
constraint, t7 after t3
constraint, t3 same-day t7
domain, t5 starts-before fri 3pm
domain, t1 ends-by fri 12pm 17
domain, t2 ends-by wed 11am 6
domain, t7 ends-by thu 9am 7
domain, t3 ends-by fri 12pm 14
domain, t5 ends-by tue 2pm 8
domain, t4 ends-by mon 9am 12
task, t1 3
task, t2 4
task, t3 4
task, t4 2
task, t5 3
task, t6 1
task, t7 3
task, t8 3
constraint, t7 before t6
constraint, t7 starts-at t2
constraint, t6 starts-at t2
domain, t2 ends-before tue 9am
domain, t5 ends-after fri 1pm
domain, t7 ends-by mon 4pm 13
domain, t3 ends-by mon 4pm 8
domain, t4 ends-by mon 12pm 16
domain, t8 ends-by tue 2pm 8
domain, t2 ends-by thu 3pm 6
domain, t5 ends-by wed 4pm 8
No (more) solutions. Total of 1 paths expanded.
task, t1 3
task, t2 4
task, t3 4
task, t4 2
task, t5 3
task, t6 1
task, t7 3
task, t8 3
constraint, t7 before t6
constraint, t7 starts-at t2
constraint, t6 starts-at t2
domain, t2 ends-before tue 9am
domain, t5 ends-after fri 1pm
domain, t7 ends-by mon 4pm 13
```

```
domain, t3 ends-by mon 4pm 8
domain, t4 ends-by mon 12pm 16
domain, t8 ends-by tue 2pm 8
domain, t2 ends-by thu 3pm 6
domain, t5 ends-by wed 4pm 8
No (more) solutions. Total of 1 paths expanded.
task, t1 2
task, t2 2
task, t3 1
task, t4 1
task, t5 3
task, t6 2
task, t7 1
task, t8 4
constraint, t5 before t3
constraint, t4 after t8
constraint, t2 same-day t8
constraint, t7 starts-at t1
constraint, t8 before t5
domain, t4 starts-after thu 9am
domain, t7 starts-after thu 10am
domain, t5 ends-by tue 10am 8
domain, t6 ends-by thu 9am 10
domain, t8 ends-by mon 2pm 20
domain, t2 ends-by fri 10am 6
domain, t7 ends-by wed 3pm 16
domain, t1 ends-by thu 1pm 16
task, t1 2
task, t2 2
task, t3 1
task, t4 1
task, t5 3
task, t6 2
task, t7 1
task, t8 4
constraint, t5 before t3
constraint, t4 after t8
constraint, t2 same-day t8
constraint, t7 starts-at t1
constraint, t8 before t5
domain, t4 starts-after thu 9am
domain, t7 starts-after thu 10am
domain, t5 ends-by tue 10am 8
domain, t6 ends-by thu 9am 10
domain, t8 ends-by mon 2pm 20
domain, t2 ends-by fri 10am 6
domain, t7 ends-by wed 3pm 16
domain, t1 ends-by thu 1pm 16
task, t1 3
task, t2 1
task, t3 2
task, t4 1
task, t5 2
task, t6 3
task, t7 4
task, t8 1
constraint, t4 same-day t2
constraint, t5 before t1
constraint, t1 same-day t3
constraint, t2 after t8
```

```
constraint, t8 same-day t3
domain, t5 ends-after fri 4pm
domain, t3 starts-before wed 2pm
domain, t5 ends-by tue 4pm 20
domain, t2 ends-by tue 9am 20
domain, t3 ends-by mon 4pm 11
domain, t4 ends-by thu 4pm 20
domain, t1 ends-by wed 2pm 14
domain, t8 ends-by mon 2pm 12
No (more) solutions. Total of 1 paths expanded.
task, t1 3
task, t2 1
task, t3 2
task, t4 1
task, t5 2
task, t6 3
task, t7 4
task, t8 1
constraint, t4 same-day t2
constraint, t5 before t1
constraint, t1 same-day t3
constraint, t2 after t8
constraint, t8 same-day t3
domain, t5 ends-after fri 4pm
domain, t3 starts-before wed 2pm
domain, t5 ends-by tue 4pm 20
domain, t2 ends-by tue 9am 20
domain, t3 ends-by mon 4pm 11
domain, t4 ends-by thu 4pm 20
domain, t1 ends-by wed 2pm 14
domain, t8 ends-by mon 2pm 12
No (more) solutions. Total of 1 paths expanded.
task, t1 4
task, t2 4
task, t3 2
task, t4 1
task, t5 1
task, t6 2
task, t7 3
task, t8 1
constraint, t3 after t2
constraint, t5 before t6
constraint, t1 same-day t7
constraint, t7 after t8
domain, t3 ends-by fri 1pm 18
domain, t5 ends-by mon 1pm 14
domain, t8 ends-by mon 1pm 8
domain, t7 ends-by fri 9am 14
task, t1 4
task, t2 4
task, t3 2
task, t4 1
task, t5 1
task, t6 2
task, t7 3
task, t8 1
constraint, t3 after t2
constraint, t5 before t6
constraint, t1 same-day t7
constraint, t7 after t8
```

```
domain, t3 ends-by fri 1pm 18
domain, t5 ends-by mon 1pm 14
domain, t8 ends-by mon 1pm 8
domain, t7 ends-by fri 9am 14
task, t1 1
task, t2 1
task, t3 4
task, t4 2
task, t5 1
task, t6 3
task, t7 3
task, t8 2
constraint, t7 starts-at t2
constraint, t1 after t5
constraint, t3 after t5
domain, t8 starts-before fri 2pm
domain, t8 starts-after thu 9am
domain, t1 ends-by thu 4pm 12
domain, t8 ends-by thu 12pm 13
domain, t3 ends-by fri 1pm 13
domain, t5 ends-by wed 10am 15
domain, t6 ends-by mon 2pm 17
domain, t7 ends-by tue 11am 7
task, t1 1
task, t2 1
task, t3 4
task, t4 2
task, t5 1
task, t6 3
task, t7 3
task, t8 2
constraint, t7 starts-at t2
constraint, t1 after t5
constraint, t3 after t5
domain, t8 starts-before fri 2pm
domain, t8 starts-after thu 9am
domain, t1 ends-by thu 4pm 12
domain, t8 ends-by thu 12pm 13
domain, t3 ends-by fri 1pm 13
domain, t5 ends-by wed 10am 15
domain, t6 ends-by mon 2pm 17
domain, t7 ends-by tue 11am 7
task, t1 2
task, t2 3
task, t3 1
task, t4 1
task, t5 4
task, t6 1
task, t7 2
task, t8 2
constraint, t8 after t1
constraint, t2 same-day t4
constraint, t7 before t1
domain, t5 ends-after tue 4pm
domain, t2 ends-by tue 3pm 19
domain, t6 ends-by thu 9am 11
domain, t4 ends-by thu 3pm 8
domain, t1 ends-by thu 9am 18
domain, t8 ends-by tue 2pm 10
task, t1 2
```

```
task, t2 3
task, t3 1
task, t4 1
task, t5 4
task, t6 1
task, t7 2
task, t8 2
constraint, t8 after t1
constraint, t2 same-day t4
constraint, t7 before t1
domain, t5 ends-after tue 4pm
domain, t2 ends-by tue 3pm 19
domain, t6 ends-by thu 9am 11
domain, t4 ends-by thu 3pm 8
domain, t1 ends-by thu 9am 18
domain, t8 ends-by tue 2pm 10
task, t1 1
task, t2 4
task, t3 2
task, t4 3
task, t5 1
task, t6 4
task, t7 4
task, t8 3
task, t9 3
task, t10 4
constraint, t3 after t9
constraint, t6 starts-at t1
constraint, t4 starts-at t2
constraint, t6 after t9
constraint, t10 after t6
domain, t8 ends-by mon 3pm 14
domain, t9 ends-by mon 10am 6
domain, t4 ends-by tue 9am 16
domain, t2 ends-by thu 9am 7
domain, t7 ends-by wed 3pm 16
task, t1 1
task, t2 4
task, t3 2
task, t4 3
task, t5 1
task, t6 4
task, t7 4
task, t8 3
task, t9 3
task, t10 4
constraint, t3 after t9
constraint, t6 starts-at t1
constraint, t4 starts-at t2
constraint, t6 after t9
constraint, t10 after t6
domain, t8 ends-by mon 3pm 14
domain, t9 ends-by mon 10am 6
domain, t4 ends-by tue 9am 16
domain, t2 ends-by thu 9am 7
domain, t7 ends-by wed 3pm 16
task, t1 1
task, t2 3
task, t3 2
task, t4 2
```

```
task, t5 1
task, t6 3
task, t7 3
task, t8 1
task, t9 2
task, t10 4
constraint, t4 before t7
constraint, t7 after t9
constraint, t1 starts-at t10
constraint, t7 starts-at t6
constraint, t4 same-day t3
domain, t8 ends-by mon 4pm 17
domain, t1 ends-by tue 4pm 17
domain, t5 ends-by fri 11am 10
domain, t7 ends-by fri 1pm 5
domain, t6 ends-by tue 4pm 11
domain, t9 ends-by wed 4pm 6
domain, t2 ends-by fri 4pm 19
domain, t4 ends-by mon 4pm 15
task, t1 1
task, t2 3
task, t3 2
task, t4 2
task, t5 1
task, t6 3
task, t7 3
task, t8 1
task, t9 2
task, t10 4
constraint, t4 before t7
constraint, t7 after t9
constraint, t1 starts-at t10
constraint, t7 starts-at t6
constraint, t4 same-day t3
domain, t8 ends-by mon 4pm 17
domain, t1 ends-by tue 4pm 17
domain, t5 ends-by fri 11am 10
domain, t7 ends-by fri 1pm 5
domain, t6 ends-by tue 4pm 11
domain, t9 ends-by wed 4pm 6
domain, t2 ends-by fri 4pm 19
domain, t4 ends-by mon 4pm 15
task, t1 4
task, t2 1
task, t3 1
task, t4 1
task, t5 1
task, t6 1
task, t7 1
task, t8 1
task, t9 3
task, t10 1
constraint, t5 same-day t6
constraint, t1 after t8
constraint, t7 same-day t2
domain, t9 ends-after thu 9am
domain, t1 ends-after mon 2pm
domain, t9 ends-by fri 10am 8
domain, t3 ends-by mon 1pm 15
domain, t1 ends-by mon 11am 10
```

```
domain, t6 ends-by thu 9am 20
domain, t2 ends-by thu 10am 9
domain, t8 ends-by tue 10am 20
task, t1 4
task, t2 1
task, t3 1
task, t4 1
task, t5 1
task, t6 1
task, t7 1
task, t8 1
task, t9 3
task, t10 1
constraint, t5 same-day t6
constraint, t1 after t8
constraint, t7 same-day t2
domain, t9 ends-after thu 9am
domain, t1 ends-after mon 2pm
domain, t9 ends-by fri 10am 8
domain, t3 ends-by mon 1pm 15
domain, t1 ends-by mon 11am 10
domain, t6 ends-by thu 9am 20
domain, t2 ends-by thu 10am 9
domain, t8 ends-by tue 10am 20
task, t1 4
task, t2 4
task, t3 1
task, t4 1
task, t5 4
task, t6 4
task, t7 3
task, t8 1
task, t9 3
task, t10 1
constraint, t2 starts-at t9
constraint, t9 starts-at t7
constraint, t3 after t9
domain, t8 starts-after tue 12pm
domain, t2 ends-by fri 12pm 11
domain, t7 ends-by fri 1pm 19
domain, t5 ends-by thu 4pm 20
domain, t6 ends-by thu 10am 8
domain, t4 ends-by mon 2pm 20
No (more) solutions. Total of 1 paths expanded.
task, t1 4
task, t2 4
task, t3 1
task, t4 1
task, t5 4
task, t6 4
task, t7 3
task, t8 1
task, t9 3
task, t10 1
constraint, t2 starts-at t9
constraint, t9 starts-at t7
constraint, t3 after t9
domain, t8 starts-after tue 12pm
domain, t2 ends-by fri 12pm 11
domain, t7 ends-by fri 1pm 19
```

```
domain, t5 ends-by thu 4pm 20
domain, t6 ends-by thu 10am 8
domain, t4 ends-by mon 2pm 20
No (more) solutions. Total of 1 paths expanded.
task, t1 2
task, t2 2
task, t3 3
task, t4 2
task, t5 2
task, t6 4
task, t7 4
task, t8 2
task, t9 3
task, t10 2
constraint, t5 before t2
constraint, t3 after t8
constraint, t5 same-day t8
domain, t6 starts-before wed 9am
domain, t10 starts-before thu 10am
domain, t3 ends-by thu 3pm 16
domain, t10 ends-by wed 2pm 8
domain, t7 ends-by tue 12pm 18
domain, t8 ends-by tue 3pm 12
domain, t1 ends-by tue 2pm 19
domain, t4 ends-by tue 3pm 14
domain, t9 ends-by thu 11am 11
task, t1 2
task, t2 2
task, t3 3
task, t4 2
task, t5 2
task, t6 4
task, t7 4
task, t8 2
task, t9 3
task, t10 2
constraint, t5 before t2
constraint, t3 after t8
constraint, t5 same-day t8
domain, t6 starts-before wed 9am
domain, t10 starts-before thu 10am
domain, t3 ends-by thu 3pm 16
domain, t10 ends-by wed 2pm 8
domain, t7 ends-by tue 12pm 18
domain, t8 ends-by tue 3pm 12
domain, t1 ends-by tue 2pm 19
domain, t4 ends-by tue 3pm 14
domain, t9 ends-by thu 11am 11
task, t1 3
task, t2 1
task, t3 4
task, t4 2
task, t5 4
task, t6 1
task, t7 2
task, t8 3
task, t9 1
task, t10 1
constraint, t3 before t1
constraint, t8 before t9
```

```
constraint, t9 after t5
constraint, t2 starts-at t1
constraint, t10 same-day t8
domain, t7 starts-after wed 11am
domain, t3 starts-before wed 3pm
domain, t1 ends-by fri 11am 14
domain, t3 ends-by thu 10am 13
domain, t2 ends-by wed 3pm 18
domain, t7 ends-by fri 1pm 12
domain, t8 ends-by thu 1pm 16
domain, t6 ends-by thu 3pm 17
domain, t5 ends-by thu 4pm 16
domain, t4 ends-by mon 2pm 7
task, t1 3
task, t2 1
task, t3 4
task, t4 2
task, t5 4
task, t6 1
task, t7 2
task, t8 3
task, t9 1
task, t10 1
constraint, t3 before t1
constraint, t8 before t9
constraint, t9 after t5
constraint, t2 starts-at t1
constraint, t10 same-day t8
domain, t7 starts-after wed 11am
domain, t3 starts-before wed 3pm
domain, t1 ends-by fri 11am 14
domain, t3 ends-by thu 10am 13
domain, t2 ends-by wed 3pm 18
domain, t7 ends-by fri 1pm 12
domain, t8 ends-by thu 1pm 16
domain, t6 ends-by thu 3pm 17
domain, t5 ends-by thu 4pm 16
domain, t4 ends-by mon 2pm 7
task, t1 4
task, t2 2
task, t3 2
task, t4 1
task, t5 3
task, t6 1
task, t7 1
task, t8 4
task, t9 2
task, t10 1
constraint, t4 same-day t10
constraint, t4 after t8
constraint, t2 same-day t6
constraint, t9 same-day t8
domain, t2 starts-after thu 11am
domain, t10 ends-by thu 2pm 6
domain, t1 ends-by fri 10am 6
domain, t8 ends-by mon 12pm 20
domain, t5 ends-by wed 4pm 20
domain, t3 ends-by thu 11am 10
domain, t2 ends-by wed 3pm 15
domain, t9 ends-by thu 12pm 12
```

```
domain, t6 ends-by fri 10am 16
task, t1 4
task, t2 2
task, t3 2
task, t4 1
task, t5 3
task, t6 1
task, t7 1
task, t8 4
task, t9 2
task, t10 1
constraint, t4 same-day t10
constraint, t4 after t8
constraint, t2 same-day t6
constraint, t9 same-day t8
domain, t2 starts-after thu 11am
domain, t10 ends-by thu 2pm 6
domain, t1 ends-by fri 10am 6
domain, t8 ends-by mon 12pm 20
domain, t5 ends-by wed 4pm 20
domain, t3 ends-by thu 11am 10
domain, t2 ends-by wed 3pm 15
domain, t9 ends-by thu 12pm 12
domain, t6 ends-by fri 10am 16
task, t1 4
task, t2 3
task, t3 4
task, t4 1
task, t5 2
task, t6 2
task, t7 2
task, t8 3
task, t9 4
task, t10 4
constraint, t7 starts-at t8
constraint, t3 starts-at t6
constraint, t4 same-day t1
constraint, t2 same-day t4
constraint, t6 same-day t2
domain, t9 ends-by tue 4pm 7
domain, t8 ends-by mon 10am 19
domain, t10 ends-by tue 9am 16
domain, t6 ends-by tue 10am 12
domain, t7 ends-by thu 1pm 19
domain, t3 ends-by tue 12pm 6
task, t1 4
task, t2 3
task, t3 4
task, t4 1
task, t5 2
task, t6 2
task, t7 2
task, t8 3
task, t9 4
task, t10 4
constraint, t7 starts-at t8
constraint, t3 starts-at t6
constraint, t4 same-day t1
constraint, t2 same-day t4
constraint, t6 same-day t2
```

```
domain, t9 ends-by tue 4pm 7
domain, t8 ends-by mon 10am 19
domain, t10 ends-by tue 9am 16
domain, t6 ends-by tue 10am 12
domain, t7 ends-by thu 1pm 19
domain, t3 ends-by tue 12pm 6
task, t1 2
task, t2 1
task, t3 4
task, t4 4
task, t5 3
task, t6 3
task, t7 4
task, t8 2
task, t9 1
task, t10 3
constraint, t4 after t8
constraint, t4 starts-at t7
constraint, t8 same-day t2
constraint, t9 same-day t7
constraint, t3 same-day t1
domain, t3 ends-by wed 4pm 14
domain, t2 ends-by tue 10am 10
domain, t6 ends-by thu 9am 12
domain, t9 ends-by thu 3pm 17
domain, t1 ends-by wed 2pm 10
No (more) solutions. Total of 1 paths expanded.
task, t1 2
task, t2 1
task, t3 4
task, t4 4
task, t5 3
task, t6 3
task, t7 4
task, t8 2
task, t9 1
task, t10 3
constraint, t4 after t8
constraint, t4 starts-at t7
constraint, t8 same-day t2
constraint, t9 same-day t7
constraint, t3 same-day t1
domain, t3 ends-by wed 4pm 14
domain, t2 ends-by tue 10am 10
domain, t6 ends-by thu 9am 12
domain, t9 ends-by thu 3pm 17
domain, t1 ends-by wed 2pm 10
No (more) solutions. Total of 1 paths expanded.
task, t1 1
task, t2 1
task, t3 1
task, t4 2
task, t5 2
task, t6 3
task, t7 4
task, t8 2
task, t9 3
task, t10 1
constraint, t8 before t7
constraint, t5 before t4
```

```
constraint, t10 same-day t9
domain, t9 ends-by mon 4pm 7
domain, t4 ends-by wed 1pm 15
domain, t8 ends-by tue 4pm 13
domain, t6 ends-by thu 12pm 8
domain, t3 ends-by thu 11am 17
domain, t5 ends-by wed 2pm 6
domain, t2 ends-by fri 11am 17
domain, t7 ends-by wed 2pm 9
task, t1 1
task, t2 1
task, t3 1
task, t4 2
task, t5 2
task, t6 3
task, t7 4
task, t8 2
task, t9 3
task, t10 1
constraint, t8 before t7
constraint, t5 before t4
constraint, t10 same-day t9
domain, t9 ends-by mon 4pm 7
domain, t4 ends-by wed 1pm 15
domain, t8 ends-by tue 4pm 13
domain, t6 ends-by thu 12pm 8
domain, t3 ends-by thu 11am 17
domain, t5 ends-by wed 2pm 6
domain, t2 ends-by fri 11am 17
domain, t7 ends-by wed 2pm 9
task, t1 4
task, t2 2
task, t3 1
task, t4 4
task, t5 3
task, t6 1
task, t7 1
task, t8 3
task, t9 4
task, t10 4
task, t11 4
task, t12 2
constraint, t2 same-day t3
constraint, t10 starts-at t11
constraint, t3 same-day t1
constraint, t4 after t5
domain, t4 starts-before wed 4pm
domain, t7 starts-after fri 10am
domain, t10 ends-before tue 4pm
domain, t9 ends-by wed 10am 9
domain, t5 ends-by thu 3pm 15
domain, t6 ends-by wed 12pm 11
domain, t8 ends-by mon 9am 8
domain, t4 ends-by tue 9am 18
domain, t1 ends-by tue 3pm 17
domain, t10 ends-by fri 11am 15
domain, t7 ends-by thu 2pm 18
No (more) solutions. Total of 1 paths expanded.
task, t1 4
task, t2 2
```

```
task, t3 1
task, t4 4
task, t5 3
task, t6 1
task, t7 1
task, t8 3
task, t9 4
task, t10 4
task, t11 4
task, t12 2
constraint, t2 same-day t3
constraint, t10 starts-at t11
constraint, t3 same-day t1
constraint, t4 after t5
domain, t4 starts-before wed 4pm
domain, t7 starts-after fri 10am
domain, t10 ends-before tue 4pm
domain, t9 ends-by wed 10am 9
domain, t5 ends-by thu 3pm 15
domain, t6 ends-by wed 12pm 11
domain, t8 ends-by mon 9am 8
domain, t4 ends-by tue 9am 18
domain, t1 ends-by tue 3pm 17
domain, t10 ends-by fri 11am 15
domain, t7 ends-by thu 2pm 18
No (more) solutions. Total of 1 paths expanded.
task, t1 4
task, t2 1
task, t3 3
task, t4 3
task, t5 1
task, t6 4
task, t7 3
task, t8 1
task, t9 2
task, t10 2
task, t11 3
task, t12 1
constraint, t11 before t8
constraint, t7 before t5
constraint, t8 starts-at t10
constraint, t10 before t2
constraint, t10 starts-at t1
constraint, t4 before t8
domain, t3 starts-after thu 3pm
domain, t10 ends-by fri 12pm 11
domain, t2 ends-by fri 12pm 19
domain, t3 ends-by fri 1pm 5
domain, t4 ends-by thu 4pm 19
domain, t7 ends-by wed 11am 6
domain, t8 ends-by mon 2pm 20
task, t1 4
task, t2 1
task, t3 3
task, t4 3
task, t5 1
task, t6 4
task, t7 3
task, t8 1
task, t9 2
```

```
task, t10 2
task, t11 3
task, t12 1
constraint, t11 before t8
constraint, t7 before t5
constraint, t8 starts-at t10
constraint, t10 before t2
constraint, t10 starts-at t1
constraint, t4 before t8
domain, t3 starts-after thu 3pm
domain, t10 ends-by fri 12pm 11
domain, t2 ends-by fri 12pm 19
domain, t3 ends-by fri 1pm 5
domain, t4 ends-by thu 4pm 19
domain, t7 ends-by wed 11am 6
domain, t8 ends-by mon 2pm 20
task, t1 1
task, t2 4
task, t3 2
task, t4 4
task, t5 2
task, t6 2
task, t7 4
task, t8 4
task, t9 3
task, t10 1
task, t11 3
task, t12 2
constraint, t12 after t2
constraint, t6 after t11
constraint, t3 starts-at t5
constraint, t2 before t12
constraint, t2 before t1
domain, t1 ends-after mon 11am
domain, t4 ends-after wed 4pm
domain, t1 ends-by tue 10am 13
domain, t12 ends-by tue 2pm 9
domain, t2 ends-by wed 9am 7
domain, t4 ends-by wed 12pm 10
domain, t11 ends-by thu 10am 12
domain, t10 ends-by wed 4pm 8
task, t1 1
task, t2 4
task, t3 2
task, t4 4
task, t5 2
task, t6 2
task, t7 4
task, t8 4
task, t9 3
task, t10 1
task, t11 3
task, t12 2
constraint, t12 after t2
constraint, t6 after t11
constraint, t3 starts-at t5
constraint, t2 before t12
constraint, t2 before t1
domain, t1 ends-after mon 11am
domain, t4 ends-after wed 4pm
```

```
domain, t1 ends-by tue 10am 13
domain, t12 ends-by tue 2pm 9
domain, t2 ends-by wed 9am 7
domain, t4 ends-by wed 12pm 10
domain, t11 ends-by thu 10am 12
domain, t10 ends-by wed 4pm 8
task, t1 1
task, t2 1
task, t3 4
task, t4 3
task, t5 1
task, t6 4
task, t7 3
task, t8 2
task, t9 2
task, t10 2
task, t11 4
task, t12 1
constraint, t12 starts-at t8
constraint, t12 before t5
constraint, t10 before t6
constraint, t11 same-day t10
constraint, t7 starts-at t3
constraint, t3 starts-at t9
constraint, t3 after t9
domain, t4 ends-after wed 11am
domain, t3 ends-before wed 10am
domain, t3 starts-before mon 9am
domain, t1 ends-by tue 3pm 20
domain, t11 ends-by thu 4pm 18
domain, t6 ends-by wed 9am 10
domain, t4 ends-by tue 11am 17
domain, t5 ends-by fri 3pm 16
domain, t2 ends-by mon 3pm 18
domain, t9 ends-by tue 12pm 19
No (more) solutions. Total of 1 paths expanded.
task, t1 1
task, t2 1
task, t3 4
task, t4 3
task, t5 1
task, t6 4
task, t7 3
task, t8 2
task, t9 2
task, t10 2
task, t11 4
task, t12 1
constraint, t12 starts-at t8
constraint, t12 before t5
constraint, t10 before t6
constraint, t11 same-day t10
constraint, t7 starts-at t3
constraint, t3 starts-at t9
constraint, t3 after t9
domain, t4 ends-after wed 11am
domain, t3 ends-before wed 10am
domain, t3 starts-before mon 9am
domain, t1 ends-by tue 3pm 20
domain, t11 ends-by thu 4pm 18
```

```
domain, t6 ends-by wed 9am 10
domain, t4 ends-by tue 11am 17
domain, t5 ends-by fri 3pm 16
domain, t2 ends-by mon 3pm 18
domain, t9 ends-by tue 12pm 19
No (more) solutions. Total of 1 paths expanded.
task, t1 2
task, t2 2
task, t3 2
task, t4 3
task, t5 2
task, t6 2
task, t7 4
task, t8 1
task, t9 4
task, t10 1
task, t11 3
task, t12 3
constraint, t3 starts-at t2
constraint, t9 same-day t3
constraint, t1 after t10
constraint, t1 before t3
constraint, t9 before t3
constraint, t4 before t6
constraint, t3 same-day t12
constraint, t8 before t1
domain, t10 ends-before mon 1pm
domain, t11 starts-before fri 9am
domain, t4 ends-by fri 9am 13
domain, t1 ends-by tue 12pm 10
domain, t8 ends-by mon 4pm 19
domain, t9 ends-by mon 1pm 12
domain, t10 ends-by wed 3pm 5
domain, t11 ends-by mon 12pm 8
domain, t7 ends-by thu 11am 19
task, t1 2
task, t2 2
task, t3 2
task, t4 3
task, t5 2
task, t6 2
task, t7 4
task, t8 1
task, t9 4
task, t10 1
task, t11 3
task, t12 3
constraint, t3 starts-at t2
constraint, t9 same-day t3
constraint, t1 after t10
constraint, t1 before t3
constraint, t9 before t3
constraint, t4 before t6
constraint, t3 same-day t12
constraint, t8 before t1
domain, t10 ends-before mon 1pm
domain, t11 starts-before fri 9am
domain, t4 ends-by fri 9am 13
domain, t1 ends-by tue 12pm 10
domain, t8 ends-by mon 4pm 19
```

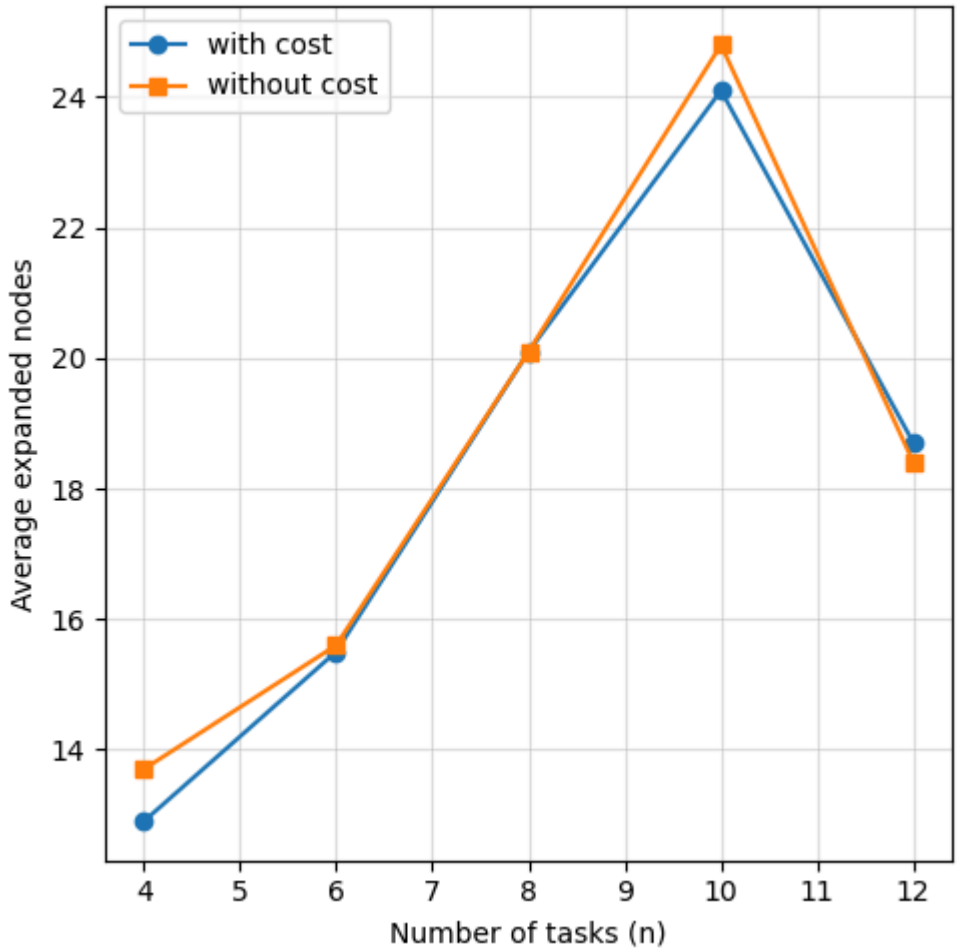
```
domain, t9 ends-by mon 1pm 12
domain, t10 ends-by wed 3pm 5
domain, t11 ends-by mon 12pm 8
domain, t7 ends-by thu 11am 19
task, t1 4
task, t2 3
task, t3 1
task, t4 4
task, t5 1
task, t6 2
task, t7 4
task, t8 1
task, t9 2
task, t10 4
task, t11 3
task, t12 2
constraint, t6 before t9
constraint, t7 starts-at t9
constraint, t1 starts-at t4
constraint, t9 same-day t2
constraint, t9 before t2
constraint, t3 starts-at t9
domain, t6 ends-after wed 3pm
domain, t2 ends-by mon 11am 11
domain, t1 ends-by fri 11am 9
domain, t8 ends-by thu 12pm 12
domain, t11 ends-by wed 2pm 5
domain, t9 ends-by mon 10am 14
domain, t7 ends-by wed 4pm 9
domain, t4 ends-by tue 9am 9
domain, t3 ends-by fri 10am 9
No (more) solutions. Total of 1 paths expanded.
task, t1 4
task, t2 3
task, t3 1
task, t4 4
task, t5 1
task, t6 2
task, t7 4
task, t8 1
task, t9 2
task, t10 4
task, t11 3
task, t12 2
constraint, t6 before t9
constraint, t7 starts-at t9
constraint, t1 starts-at t4
constraint, t9 same-day t2
constraint, t9 before t2
constraint, t3 starts-at t9
domain, t6 ends-after wed 3pm
domain, t2 ends-by mon 11am 11
domain, t1 ends-by fri 11am 9
domain, t8 ends-by thu 12pm 12
domain, t11 ends-by wed 2pm 5
domain, t9 ends-by mon 10am 14
domain, t7 ends-by wed 4pm 9
domain, t4 ends-by tue 9am 9
domain, t3 ends-by fri 10am 9
No (more) solutions. Total of 1 paths expanded.
```

```
task, t1 3
task, t2 1
task, t3 2
task, t4 2
task, t5 2
task, t6 3
task, t7 4
task, t8 4
task, t9 2
task, t10 1
task, t11 4
task, t12 1
constraint, t6 starts-at t9
constraint, t8 before t9
constraint, t3 starts-at t4
constraint, t9 after t1
constraint, t3 before t12
domain, t3 starts-after wed 9am
domain, t5 starts-after mon 11am
domain, t1 starts-after tue 10am
domain, t2 ends-by wed 12pm 10
domain, t4 ends-by thu 1pm 6
domain, t9 ends-by fri 2pm 9
domain, t1 ends-by fri 3pm 10
domain, t10 ends-by tue 2pm 13
domain, t3 ends-by mon 11am 18
domain, t8 ends-by thu 1pm 20
domain, t7 ends-by fri 10am 17
domain, t11 ends-by wed 3pm 6
task, t1 3
task, t2 1
task, t3 2
task, t4 2
task, t5 2
task, t6 3
task, t7 4
task, t8 4
task, t9 2
task, t10 1
task, t11 4
task, t12 1
constraint, t6 starts-at t9
constraint, t8 before t9
constraint, t3 starts-at t4
constraint, t9 after t1
constraint, t3 before t12
domain, t3 starts-after wed 9am
domain, t5 starts-after mon 11am
domain, t1 starts-after tue 10am
domain, t2 ends-by wed 12pm 10
domain, t4 ends-by thu 1pm 6
domain, t9 ends-by fri 2pm 9
domain, t1 ends-by fri 3pm 10
domain, t10 ends-by tue 2pm 13
domain, t3 ends-by mon 11am 18
domain, t8 ends-by thu 1pm 20
domain, t7 ends-by fri 10am 17
domain, t11 ends-by wed 3pm 6
task, t1 4
task, t2 4
```

```
task, t3 2
task, t4 1
task, t5 1
task, t6 4
task, t7 1
task, t8 3
task, t9 2
task, t10 1
task, t11 3
task, t12 3
constraint, t7 before t9
constraint, t2 same-day t5
constraint, t4 same-day t6
constraint, t8 after t12
constraint, t1 starts-at t10
constraint, t9 after t11
constraint, t11 same-day t5
constraint, t9 same-day t10
domain, t9 starts-before wed 2pm
domain, t4 ends-by fri 9am 17
domain, t5 ends-by tue 4pm 13
domain, t12 ends-by thu 12pm 12
domain, t11 ends-by mon 4pm 9
domain, t10 ends-by thu 2pm 9
domain, t9 ends-by tue 1pm 14
domain, t2 ends-by thu 10am 16
domain, t6 ends-by thu 1pm 20
domain, t3 ends-by mon 2pm 15
task, t1 4
task, t2 4
task, t3 2
task, t4 1
task, t5 1
task, t6 4
task, t7 1
task, t8 3
task, t9 2
task, t10 1
task, t11 3
task, t12 3
constraint, t7 before t9
constraint, t2 same-day t5
constraint, t4 same-day t6
constraint, t8 after t12
constraint, t1 starts-at t10
constraint, t9 after t11
constraint, t11 same-day t5
constraint, t9 same-day t10
domain, t9 starts-before wed 2pm
domain, t4 ends-by fri 9am 17
domain, t5 ends-by tue 4pm 13
domain, t12 ends-by thu 12pm 12
domain, t11 ends-by mon 4pm 9
domain, t10 ends-by thu 2pm 9
domain, t9 ends-by tue 1pm 14
domain, t2 ends-by thu 10am 16
domain, t6 ends-by thu 1pm 20
domain, t3 ends-by mon 2pm 15
task, t1 3
task, t2 4
```

```
task, t3 4
task, t4 3
task, t5 3
task, t6 2
task, t7 2
task, t8 2
task, t9 4
task, t10 3
task, t11 4
task, t12 4
constraint, t3 starts-at t9
constraint, t5 after t8
constraint, t9 starts-at t8
constraint, t4 starts-at t1
constraint, t3 before t4
constraint, t9 starts-at t5
domain, t7 ends-by fri 3pm 10
domain, t1 ends-by mon 4pm 20
domain, t2 ends-by mon 4pm 11
domain, t11 ends-by tue 1pm 20
domain, t5 ends-by fri 3pm 5
domain, t10 ends-by thu 12pm 15
No (more) solutions. Total of 1 paths expanded.
task, t1 3
task, t2 4
task, t3 4
task, t4 3
task, t5 3
task, t6 2
task, t7 2
task, t8 2
task, t9 4
task, t10 3
task, t11 4
task, t12 4
constraint, t3 starts-at t9
constraint, t5 after t8
constraint, t9 starts-at t8
constraint, t4 starts-at t1
constraint, t3 before t4
constraint, t9 starts-at t5
domain, t7 ends-by fri 3pm 10
domain, t1 ends-by mon 4pm 20
domain, t2 ends-by mon 4pm 11
domain, t11 ends-by tue 1pm 20
domain, t5 ends-by fri 3pm 5
domain, t10 ends-by thu 12pm 15
No (more) solutions. Total of 1 paths expanded.
task, t1 2
task, t2 3
task, t3 4
task, t4 3
task, t5 1
task, t6 4
task, t7 4
task, t8 4
task, t9 3
task, t10 4
task, t11 3
task, t12 3
```

```
constraint, t3 after t1
constraint, t2 after t11
constraint, t1 starts-at t2
constraint, t4 starts-at t9
constraint, t10 before t2
constraint, t11 starts-at t8
constraint, t6 before t10
constraint, t10 after t8
domain, t2 ends-by thu 3pm 7
domain, t3 ends-by mon 2pm 16
domain, t9 ends-by tue 4pm 9
domain, t6 ends-by wed 3pm 12
domain, t8 ends-by mon 4pm 10
domain, t10 ends-by wed 2pm 19
domain, t7 ends-by wed 12pm 17
task, t1 2
task, t2 3
task, t3 4
task, t4 3
task, t5 1
task, t6 4
task, t7 4
task, t8 4
task, t9 3
task, t10 4
task, t11 3
task, t12 3
constraint, t3 after t1
constraint, t2 after t11
constraint, t1 starts-at t2
constraint, t4 starts-at t9
constraint, t10 before t2
constraint, t11 starts-at t8
constraint, t6 before t10
constraint, t10 after t8
domain, t2 ends-by thu 3pm 7
domain, t3 ends-by mon 2pm 16
domain, t9 ends-by tue 4pm 9
domain, t6 ends-by wed 3pm 12
domain, t8 ends-by mon 4pm 10
domain, t10 ends-by wed 2pm 19
domain, t7 ends-by wed 12pm 17
  task_num |    with_cost | without_cost |  gain %
-------------------------------------------
         4 |         12.9 |         13.7 |    5.8%
         6 |         15.5 |         15.6 |    0.6%
         8 |         20.1 |         20.1 |    0.0%
        10 |         24.1 |         24.8 |    2.8%
        12 |         18.7 |         18.4 |   -1.6%
```



In [131…

```python
# Q4 extra code
def statistical_data_q4():
    ns=(2,4,6)
    trials=5
    rows = []
    rng = random.Random(0)
    for n in ns:
        dfs_sum = mrv_sum = 0
        dfs_cost_sum = mrv_cost_sum =  0
        for _ in range(trials):
            spec = generate_problem(n, seed=rng.randrange(0,1000))
            d, m, dc, mc = dfs_vs_mrv(spec)
            dfs_sum += d
            mrv_sum += m
            dfs_cost_sum += dc
            mrv_cost_sum += mc
        dfs_avg = dfs_sum / trials
        mrv_avg = mrv_sum / trials
        avg_dfs_cost = dfs_cost_sum / trials
        avg_mrv_cost = mrv_cost_sum / trials
        speedup = (dfs_avg / mrv_avg) if mrv_avg > 0 else float('inf')
        rows.append({
            "n": n,
            "avg_dfs": round(dfs_avg, 1),
            "avg_mrv": round(mrv_avg, 1),
            "dfs_cost": round(avg_dfs_cost,1),
            "mrv_cost": round(avg_mrv_cost,1),
            "speedup": round(speedup, 2),
        })
    return rows
```

```
def print_table_q4(rows):
    print(f'{"n":>4} | {"DFS avg":>10} | {"DFS-MRV avg":>12} | {"DFS_cost":>8}|
    print('-' * 65)
    for r in rows:
        print(f'{r["n"]:>4} | {r["avg_dfs"]:>10} | {r["avg_mrv"]:>12} | {r["dfs_

rows = statistical_data_q4()
print_table_q4(rows)
```

```
task, t1 4
task, t2 3
constraint, t1 same-day t2
domain, t2 ends-by tue 11am 10
domain, t1 ends-by fri 1pm 9
task, t1 4
task, t2 3
constraint, t1 same-day t2
domain, t2 ends-by tue 11am 10
domain, t1 ends-by fri 1pm 9
Nodes expanded to reach solution: 2
Nodes expanded to reach solution: 2
task, t1 2
task, t2 1
constraint, t2 after t1
domain, t1 ends-by fri 1pm 6
task, t1 2
task, t2 1
constraint, t2 after t1
domain, t1 ends-by fri 1pm 6
Nodes expanded to reach solution: 4
Nodes expanded to reach solution: 2
task, t1 2
task, t2 3
constraint, t2 starts-at t1
domain, t1 ends-by wed 3pm 9
task, t1 2
task, t2 3
constraint, t2 starts-at t1
domain, t1 ends-by wed 3pm 9
Nodes expanded to reach solution: 4
Nodes expanded to reach solution: 4
task, t1 2
task, t2 4
constraint, t2 after t1
domain, t1 ends-by mon 11am 6
task, t1 2
task, t2 4
constraint, t2 after t1
domain, t1 ends-by mon 11am 6
Nodes expanded to reach solution: 4
Nodes expanded to reach solution: 4
task, t1 3
task, t2 3
domain, t1 ends-by fri 11am 16
task, t1 3
task, t2 3
domain, t1 ends-by fri 11am 16
Nodes expanded to reach solution: 2
Nodes expanded to reach solution: 2
task, t1 1
task, t2 1
task, t3 1
task, t4 4
constraint, t4 after t3
constraint, t3 after t4
domain, t4 ends-by mon 4pm 15
domain, t3 ends-by thu 10am 16
domain, t1 ends-by wed 4pm 17
task, t1 1
```

```
task, t2 1
task, t3 1
task, t4 4
constraint, t4 after t3
constraint, t3 after t4
domain, t4 ends-by mon 4pm 15
domain, t3 ends-by thu 10am 16
domain, t1 ends-by wed 4pm 17
task, t1 3
task, t2 1
task, t3 2
task, t4 3
constraint, t3 before t4
constraint, t4 before t3
domain, t4 starts-before wed 12pm
domain, t3 ends-by mon 12pm 9
domain, t2 ends-by wed 10am 5
domain, t4 ends-by fri 2pm 15
task, t1 3
task, t2 1
task, t3 2
task, t4 3
constraint, t3 before t4
constraint, t4 before t3
domain, t4 starts-before wed 12pm
domain, t3 ends-by mon 12pm 9
domain, t2 ends-by wed 10am 5
domain, t4 ends-by fri 2pm 15
task, t1 4
task, t2 4
task, t3 2
task, t4 2
constraint, t3 after t2
constraint, t2 before t3
domain, t4 starts-before mon 2pm
domain, t2 ends-by tue 11am 8
domain, t4 ends-by thu 10am 19
task, t1 4
task, t2 4
task, t3 2
task, t4 2
constraint, t3 after t2
constraint, t2 before t3
domain, t4 starts-before mon 2pm
domain, t2 ends-by tue 11am 8
domain, t4 ends-by thu 10am 19
Nodes expanded to reach solution: 8
Nodes expanded to reach solution: 4
task, t1 1
task, t2 3
task, t3 4
task, t4 2
constraint, t4 same-day t2
domain, t1 ends-after thu 10am
domain, t1 ends-by wed 4pm 11
domain, t4 ends-by mon 9am 16
task, t1 1
task, t2 3
task, t3 4
task, t4 2
```

```
constraint, t4 same-day t2
domain, t1 ends-after thu 10am
domain, t1 ends-by wed 4pm 11
domain, t4 ends-by mon 9am 16
Nodes expanded to reach solution: 4
Nodes expanded to reach solution: 4
task, t1 4
task, t2 4
task, t3 2
task, t4 3
constraint, t1 same-day t2
domain, t1 ends-after fri 4pm
domain, t4 ends-by wed 4pm 6
domain, t3 ends-by mon 3pm 6
task, t1 4
task, t2 4
task, t3 2
task, t4 3
constraint, t1 same-day t2
domain, t1 ends-after fri 4pm
domain, t4 ends-by wed 4pm 6
domain, t3 ends-by mon 3pm 6
Nodes expanded to reach solution: 20
Nodes expanded to reach solution: 4
task, t1 3
task, t2 4
task, t3 4
task, t4 2
task, t5 4
task, t6 2
constraint, t2 same-day t3
constraint, t6 before t2
constraint, t3 before t1
domain, t3 starts-after tue 11am
domain, t6 ends-by thu 11am 18
domain, t3 ends-by thu 10am 7
domain, t5 ends-by mon 3pm 7
task, t1 3
task, t2 4
task, t3 4
task, t4 2
task, t5 4
task, t6 2
constraint, t2 same-day t3
constraint, t6 before t2
constraint, t3 before t1
domain, t3 starts-after tue 11am
domain, t6 ends-by thu 11am 18
domain, t3 ends-by thu 10am 7
domain, t5 ends-by mon 3pm 7
Nodes expanded to reach solution: 3076
Nodes expanded to reach solution: 6
task, t1 3
task, t2 2
task, t3 3
task, t4 2
task, t5 1
task, t6 3
constraint, t6 after t3
constraint, t6 starts-at t2
```

```
constraint, t4 starts-at t5
domain, t5 ends-by thu 2pm 7
domain, t6 ends-by fri 9am 10
domain, t1 ends-by wed 11am 8
task, t1 3
task, t2 2
task, t3 3
task, t4 2
task, t5 1
task, t6 3
constraint, t6 after t3
constraint, t6 starts-at t2
constraint, t4 starts-at t5
domain, t5 ends-by thu 2pm 7
domain, t6 ends-by fri 9am 10
domain, t1 ends-by wed 11am 8
Nodes expanded to reach solution: 42696
Nodes expanded to reach solution: 7
task, t1 3
task, t2 2
task, t3 1
task, t4 3
task, t5 3
task, t6 3
constraint, t2 after t1
constraint, t4 before t1
constraint, t3 before t4
constraint, t5 before t2
domain, t1 ends-by fri 10am 5
domain, t5 ends-by fri 2pm 17
domain, t4 ends-by mon 2pm 6
domain, t6 ends-by tue 10am 14
task, t1 3
task, t2 2
task, t3 1
task, t4 3
task, t5 3
task, t6 3
constraint, t2 after t1
constraint, t4 before t1
constraint, t3 before t4
constraint, t5 before t2
domain, t1 ends-by fri 10am 5
domain, t5 ends-by fri 2pm 17
domain, t4 ends-by mon 2pm 6
domain, t6 ends-by tue 10am 14
Nodes expanded to reach solution: 92957
Nodes expanded to reach solution: 12
task, t1 1
task, t2 4
task, t3 2
task, t4 1
task, t5 4
task, t6 1
constraint, t1 before t5
constraint, t3 same-day t2
constraint, t1 starts-at t2
domain, t3 ends-by tue 2pm 9
domain, t2 ends-by mon 9am 17
domain, t1 ends-by thu 1pm 6
```

```
task, t1 1
task, t2 4
task, t3 2
task, t4 1
task, t5 4
task, t6 1
constraint, t1 before t5
constraint, t3 same-day t2
constraint, t1 starts-at t2
domain, t3 ends-by tue 2pm 9
domain, t2 ends-by mon 9am 17
domain, t1 ends-by thu 1pm 6
Nodes expanded to reach solution: 94
Nodes expanded to reach solution: 6
task, t1 2
task, t2 1
task, t3 2
task, t4 4
task, t5 3
task, t6 3
constraint, t1 starts-at t4
constraint, t6 starts-at t2
domain, t3 ends-by thu 9am 15
domain, t5 ends-by fri 4pm 17
domain, t1 ends-by fri 10am 7
task, t1 2
task, t2 1
task, t3 2
task, t4 4
task, t5 3
task, t6 3
constraint, t1 starts-at t4
constraint, t6 starts-at t2
domain, t3 ends-by thu 9am 15
domain, t5 ends-by fri 4pm 17
domain, t1 ends-by fri 10am 7
Nodes expanded to reach solution: 88351
Nodes expanded to reach solution: 7
   n |    DFS avg |  DFS-MRV avg | DFS_cost| MRV_cost|  speedup
-----------------------------------------------------------------
   2 |        3.2 |          2.8 |      0.0|      0.0|    1.14
   4 |   259263.4 |          9.2 |      inf|      inf| 28180.8
   6 |    45434.8 |          7.6 |     13.6|     13.6| 5978.26
```

In [130…

```python
# Q5 extra code
def statistical_data_q5():
    ns=(4, 6, 8)
    trials=1
    rng = random.Random(0)
    rows = []
    for n in ns:
        dfs_nodes, cost_nodes = [], []
        for _ in range(trials):
            spec = generate_problem(n, seed=rng.randrange(1000))
            csp1 = create_CSP_from_spec(spec)
            _ = dfs_solve1(csp1)
            dfs_nodes.append(num_expanded)
            csp2 = create_CSP_from_spec(spec)
            _ = cost_dfs_solve1(csp2)
            cost_nodes.append(num_expanded)
```

```
        avg_dfs  = sum(dfs_nodes)/trials
        avg_cost = sum(cost_nodes)/trials
        gain_pct = 0.0 if avg_dfs == 0 else (avg_dfs - avg_cost) / avg_dfs * 100

        rows.append({
            "n": n,
            "avg_dfs": round(avg_dfs, 1),
            "avg_cost": round(avg_cost, 1),
            "gain_%": round(gain_pct, 1),
        })
    return rows

def print_table_q5(rows):
    print(f'{"n":>4} | {"DFS avg":>10} | {"DFS+cost avg":>12} | {"gain %":>7}')
    print('-' * 46)
    for r in rows:
        print(f'{r["n"]:>4} | {r["avg_dfs"]:>10} | {r["avg_cost"]:>12} | {r["gai
rows_q5 = statistical_data_q5()
print_table_q5(rows_q5)
```

```
task, t1 4
task, t2 2
task, t3 1
task, t4 2
constraint, t2 same-day t1
domain, t1 ends-by wed 12pm 14
domain, t4 ends-by thu 11am 10
Nodes expanded to reach solution: 4
task, t1 4
task, t2 2
task, t3 1
task, t4 2
constraint, t2 same-day t1
domain, t1 ends-by wed 12pm 14
domain, t4 ends-by thu 11am 10
Nodes expanded to reach solution: 4
task, t1 1
task, t2 2
task, t3 2
task, t4 4
task, t5 4
task, t6 2
constraint, t4 after t3
constraint, t6 before t3
domain, t5 ends-after wed 4pm
domain, t5 ends-by mon 2pm 6
domain, t6 ends-by wed 11am 11
domain, t4 ends-by tue 11am 8
Nodes expanded to reach solution: 9817
task, t1 1
task, t2 2
task, t3 2
task, t4 4
task, t5 4
task, t6 2
constraint, t4 after t3
constraint, t6 before t3
domain, t5 ends-after wed 4pm
domain, t5 ends-by mon 2pm 6
domain, t6 ends-by wed 11am 11
domain, t4 ends-by tue 11am 8
Nodes expanded to reach solution: 6500
task, t1 2
task, t2 2
task, t3 4
task, t4 4
task, t5 2
task, t6 2
task, t7 2
task, t8 3
constraint, t2 same-day t7
constraint, t4 before t2
constraint, t3 same-day t6
constraint, t3 starts-at t8
constraint, t3 starts-at t6
domain, t4 ends-by wed 11am 15
domain, t7 ends-by wed 9am 13
domain, t1 ends-by mon 1pm 9
domain, t8 ends-by wed 4pm 16
Nodes expanded to reach solution: 9946
```

```
task, t1 2
task, t2 2
task, t3 4
task, t4 4
task, t5 2
task, t6 2
task, t7 2
task, t8 3
constraint, t2 same-day t7
constraint, t4 before t2
constraint, t3 same-day t6
constraint, t3 starts-at t8
constraint, t3 starts-at t6
domain, t4 ends-by wed 11am 15
domain, t7 ends-by wed 9am 13
domain, t1 ends-by mon 1pm 9
domain, t8 ends-by wed 4pm 16
Nodes expanded to reach solution: 1617
   n |    DFS avg | DFS+cost avg |  gain %
---------------------------------------------
   4 |        4.0 |          4.0 |    0.0%
   6 |     9817.0 |       6500.0 |   33.8%
   8 |     9946.0 |       1617.0 |   83.7%
```

In [ ]:

```python
# Q6 extra code
import random
import matplotlib.pyplot as plt

def slove_k_split(spec, k):
    csp = create_CSP_from_spec(spec)
    solver = Search_with_AC_from_Cost_CSP_Ksplit(csp, k=k)
    nei = solver.neighbors(solver.start_node())
    return solver.nodes_expend

def plot_neighbors_vs_k_across_n():
    ns=(4,6,8,10)
    ks=(2,4,6,8)
    trials=10
    seed_base=0
    rng = random.Random(seed_base)
    ks_list = list(ks)
    plt.figure()
    for n in ns:
        avg_nodes_for_n = []
        for k in ks_list:
            total = 0
            for _ in range(trials):
                spec = generate_problem(n, seed=rng.randrange(1000))
                total += slove_k_split(spec, k)
            avg_nodes_for_n.append(total / trials)
        plt.plot(ks_list, avg_nodes_for_n, marker='o', label=f'n={n}')

    plt.xlabel('k (number of domain partitions)')
    plt.ylabel('Average nodes_expend')
    plt.title('k-way Splitting')
    plt.grid(True)
    plt.legend()
    plt.show()

plot_neighbors_vs_k_across_n()
```

```
task, t1 1
task, t2 1
task, t3 3
task, t4 2
constraint, t2 starts-at t4
constraint, t4 same-day t1
domain, t2 ends-after mon 4pm
domain, t2 ends-by mon 10am 12
domain, t1 ends-by tue 9am 15
task, t1 4
task, t2 3
task, t3 2
task, t4 2
constraint, t2 same-day t3
constraint, t2 after t4
domain, t2 ends-after fri 2pm
domain, t1 ends-by fri 11am 12
domain, t4 ends-by mon 3pm 20
task, t1 4
task, t2 3
task, t3 1
task, t4 2
constraint, t1 same-day t3
domain, t2 ends-by tue 2pm 6
domain, t1 ends-by wed 10am 10
domain, t4 ends-by mon 10am 14
task, t1 1
task, t2 2
task, t3 2
task, t4 3
constraint, t4 after t3
constraint, t4 before t3
domain, t2 starts-after tue 3pm
domain, t3 ends-by tue 1pm 16
domain, t2 ends-by wed 10am 11
domain, t1 ends-by mon 10am 8
task, t1 2
task, t2 3
task, t3 3
task, t4 2
constraint, t2 before t4
domain, t4 ends-by wed 4pm 14
domain, t3 ends-by fri 4pm 6
domain, t2 ends-by mon 10am 17
task, t1 4
task, t2 3
task, t3 1
task, t4 4
constraint, t1 same-day t2
constraint, t4 same-day t2
domain, t2 ends-by thu 1pm 18
domain, t3 ends-by fri 4pm 9
task, t1 4
task, t2 3
task, t3 3
task, t4 3
constraint, t4 after t1
constraint, t1 same-day t4
domain, t2 starts-after thu 10am
domain, t1 ends-by mon 9am 11
```

```
domain, t2 ends-by wed 11am 6
domain, t4 ends-by mon 12pm 6
task, t1 4
task, t2 4
task, t3 1
task, t4 3
constraint, t1 after t4
constraint, t1 before t3
domain, t4 ends-after thu 1pm
domain, t3 ends-by mon 9am 6
domain, t2 ends-by tue 1pm 16
task, t1 2
task, t2 3
task, t3 1
task, t4 3
constraint, t1 before t2
domain, t2 ends-by tue 2pm 10
domain, t1 ends-by fri 3pm 7
task, t1 3
task, t2 4
task, t3 3
task, t4 2
constraint, t1 same-day t2
constraint, t3 before t4
domain, t1 ends-by tue 9am 6
domain, t3 ends-by fri 11am 8
task, t1 3
task, t2 1
task, t3 2
task, t4 1
constraint, t2 same-day t4
constraint, t3 starts-at t1
domain, t4 starts-before mon 10am
domain, t1 ends-by thu 10am 17
domain, t2 ends-by fri 2pm 8
domain, t3 ends-by wed 3pm 5
task, t1 2
task, t2 2
task, t3 2
task, t4 4
constraint, t4 starts-at t2
constraint, t3 after t4
domain, t1 starts-after mon 2pm
domain, t1 ends-by wed 11am 19
domain, t2 ends-by fri 1pm 6
domain, t4 ends-by mon 9am 5
task, t1 1
task, t2 1
task, t3 3
task, t4 4
constraint, t3 after t2
constraint, t4 before t2
domain, t3 starts-before mon 4pm
domain, t3 ends-by wed 12pm 20
domain, t1 ends-by fri 9am 19
task, t1 4
task, t2 4
task, t3 1
task, t4 4
constraint, t3 before t4
```

```
constraint, t4 starts-at t3
domain, t1 ends-after wed 2pm
domain, t4 ends-by fri 12pm 6
domain, t1 ends-by tue 9am 6
domain, t2 ends-by thu 12pm 18
task, t1 3
task, t2 4
task, t3 4
task, t4 3
constraint, t4 before t1
constraint, t2 after t4
domain, t3 ends-by wed 3pm 15
domain, t1 ends-by thu 1pm 14
task, t1 4
task, t2 3
task, t3 2
task, t4 4
constraint, t4 same-day t1
constraint, t1 starts-at t4
domain, t3 ends-by fri 2pm 6
domain, t1 ends-by wed 12pm 6
domain, t2 ends-by fri 3pm 10
task, t1 4
task, t2 3
task, t3 3
task, t4 2
constraint, t3 starts-at t4
domain, t4 ends-by tue 2pm 8
domain, t1 ends-by wed 11am 5
domain, t3 ends-by mon 1pm 9
task, t1 4
task, t2 4
task, t3 1
task, t4 1
constraint, t3 same-day t4
constraint, t3 starts-at t1
domain, t4 ends-after fri 3pm
domain, t2 ends-by wed 4pm 6
domain, t1 ends-by mon 10am 10
domain, t4 ends-by mon 1pm 15
task, t1 1
task, t2 4
task, t3 4
task, t4 1
constraint, t2 before t1
constraint, t1 same-day t4
domain, t2 ends-by thu 11am 17
domain, t1 ends-by thu 9am 13
domain, t3 ends-by wed 2pm 15
task, t1 3
task, t2 2
task, t3 4
task, t4 1
constraint, t1 starts-at t4
constraint, t2 same-day t4
domain, t2 ends-by wed 12pm 14
domain, t3 ends-by tue 1pm 10
domain, t1 ends-by mon 9am 8
task, t1 3
task, t2 3
```

```
task, t3 3
task, t4 3
constraint, t1 before t4
domain, t4 ends-by tue 9am 14
domain, t3 ends-by thu 9am 15
domain, t1 ends-by tue 2pm 15
task, t1 4
task, t2 3
task, t3 2
task, t4 4
constraint, t1 same-day t3
constraint, t1 after t3
domain, t1 ends-after tue 2pm
domain, t4 ends-by mon 4pm 7
domain, t3 ends-by tue 10am 12
task, t1 2
task, t2 2
task, t3 4
task, t4 1
constraint, t1 starts-at t4
domain, t3 ends-by tue 12pm 16
domain, t4 ends-by tue 10am 10
domain, t2 ends-by tue 1pm 11
task, t1 2
task, t2 4
task, t3 2
task, t4 4
constraint, t2 before t4
domain, t4 starts-after thu 3pm
domain, t2 ends-by thu 10am 17
domain, t4 ends-by wed 1pm 10
domain, t3 ends-by fri 9am 9
task, t1 4
task, t2 1
task, t3 1
task, t4 2
constraint, t3 starts-at t2
constraint, t2 before t4
domain, t4 ends-by mon 2pm 15
domain, t3 ends-by tue 2pm 20
task, t1 2
task, t2 4
task, t3 3
task, t4 1
constraint, t4 same-day t3
domain, t3 ends-by fri 2pm 16
domain, t4 ends-by thu 10am 6
domain, t2 ends-by tue 4pm 17
task, t1 3
task, t2 1
task, t3 3
task, t4 3
constraint, t3 before t1
constraint, t2 starts-at t1
domain, t4 ends-by wed 4pm 13
domain, t1 ends-by mon 3pm 6
domain, t2 ends-by mon 3pm 12
task, t1 3
task, t2 3
task, t3 2
```

```
task, t4 2
constraint, t3 after t1
domain, t2 ends-before mon 11am
domain, t2 ends-by thu 1pm 9
domain, t1 ends-by tue 4pm 6
task, t1 2
task, t2 2
task, t3 1
task, t4 4
constraint, t2 starts-at t1
constraint, t3 same-day t2
domain, t2 starts-after fri 11am
domain, t1 ends-by fri 2pm 13
domain, t4 ends-by mon 4pm 13
domain, t2 ends-by mon 1pm 5
task, t1 1
task, t2 2
task, t3 3
task, t4 1
constraint, t1 before t4
constraint, t2 after t4
domain, t2 ends-by thu 9am 18
domain, t3 ends-by thu 10am 13
domain, t4 ends-by wed 11am 7
task, t1 1
task, t2 2
task, t3 4
task, t4 4
constraint, t1 before t2
constraint, t4 before t1
domain, t4 ends-by fri 3pm 8
domain, t1 ends-by wed 3pm 10
domain, t3 ends-by thu 10am 16
task, t1 3
task, t2 4
task, t3 3
task, t4 2
constraint, t4 starts-at t3
constraint, t4 after t2
domain, t4 starts-after fri 11am
domain, t1 ends-by tue 4pm 19
domain, t3 ends-by wed 12pm 16
task, t1 1
task, t2 1
task, t3 3
task, t4 3
constraint, t2 same-day t3
constraint, t4 starts-at t2
domain, t1 ends-by thu 9am 17
domain, t2 ends-by mon 10am 12
domain, t4 ends-by thu 10am 7
task, t1 2
task, t2 1
task, t3 4
task, t4 1
constraint, t2 same-day t3
domain, t4 starts-before thu 12pm
domain, t1 ends-by tue 12pm 6
domain, t2 ends-by tue 9am 13
task, t1 4
```

```
task, t2 1
task, t3 3
task, t4 2
constraint, t3 starts-at t4
domain, t3 ends-by mon 12pm 19
domain, t4 ends-by thu 3pm 16
task, t1 2
task, t2 2
task, t3 1
task, t4 2
constraint, t1 starts-at t3
domain, t3 ends-by mon 4pm 10
domain, t1 ends-by wed 1pm 17
task, t1 1
task, t2 3
task, t3 3
task, t4 1
constraint, t1 starts-at t4
constraint, t4 starts-at t1
domain, t2 ends-before mon 2pm
domain, t3 ends-by thu 10am 5
domain, t2 ends-by tue 1pm 12
task, t1 4
task, t2 1
task, t3 2
task, t4 1
constraint, t3 starts-at t4
domain, t4 ends-after tue 9am
domain, t4 ends-by mon 3pm 12
domain, t3 ends-by fri 10am 10
task, t1 1
task, t2 4
task, t3 1
task, t4 1
constraint, t1 before t3
constraint, t3 before t1
domain, t4 ends-by wed 12pm 18
domain, t1 ends-by thu 2pm 15
task, t1 3
task, t2 2
task, t3 3
task, t4 3
constraint, t2 before t1
domain, t2 starts-before fri 11am
domain, t2 ends-by thu 11am 13
domain, t4 ends-by thu 12pm 8
domain, t3 ends-by mon 11am 18
task, t1 1
task, t2 1
task, t3 2
task, t4 4
task, t5 1
task, t6 3
constraint, t4 before t2
constraint, t1 same-day t5
constraint, t2 before t4
domain, t5 ends-by fri 4pm 14
domain, t4 ends-by tue 11am 16
domain, t1 ends-by thu 1pm 11
task, t1 2
```

```
task, t2 3
task, t3 1
task, t4 2
task, t5 3
task, t6 3
constraint, t5 same-day t3
constraint, t1 before t5
constraint, t1 before t4
domain, t1 ends-by thu 9am 20
domain, t6 ends-by thu 2pm 20
domain, t4 ends-by tue 12pm 12
domain, t2 ends-by mon 12pm 17
task, t1 3
task, t2 2
task, t3 3
task, t4 2
task, t5 3
task, t6 1
constraint, t5 same-day t1
constraint, t2 same-day t6
constraint, t1 after t5
constraint, t1 starts-at t3
domain, t1 ends-by thu 10am 15
domain, t3 ends-by wed 1pm 18
domain, t5 ends-by fri 12pm 13
domain, t2 ends-by mon 2pm 18
task, t1 4
task, t2 3
task, t3 3
task, t4 3
task, t5 3
task, t6 3
constraint, t2 after t5
constraint, t6 before t2
constraint, t6 after t1
domain, t6 starts-after thu 4pm
domain, t6 ends-by tue 9am 18
domain, t1 ends-by thu 1pm 12
domain, t2 ends-by tue 11am 11
task, t1 2
task, t2 2
task, t3 4
task, t4 3
task, t5 3
task, t6 4
constraint, t5 same-day t3
constraint, t4 starts-at t3
domain, t3 ends-by wed 4pm 19
domain, t4 ends-by tue 9am 19
domain, t1 ends-by thu 12pm 17
task, t1 2
task, t2 1
task, t3 1
task, t4 4
task, t5 1
task, t6 4
constraint, t3 starts-at t1
constraint, t6 after t5
constraint, t2 after t5
constraint, t4 after t6
```

```
domain, t6 ends-by wed 10am 7
domain, t5 ends-by tue 1pm 15
domain, t4 ends-by wed 4pm 7
domain, t3 ends-by tue 10am 17
task, t1 3
task, t2 3
task, t3 3
task, t4 2
task, t5 1
task, t6 2
constraint, t5 same-day t1
constraint, t4 same-day t2
domain, t1 ends-by wed 9am 5
domain, t3 ends-by tue 1pm 16
domain, t4 ends-by wed 4pm 17
domain, t5 ends-by wed 4pm 5
task, t1 2
task, t2 2
task, t3 4
task, t4 2
task, t5 3
task, t6 2
constraint, t6 after t5
constraint, t2 starts-at t3
constraint, t1 before t5
constraint, t2 same-day t3
domain, t2 ends-by fri 11am 11
domain, t6 ends-by fri 3pm 18
domain, t3 ends-by mon 1pm 10
task, t1 4
task, t2 3
task, t3 4
task, t4 4
task, t5 2
task, t6 2
constraint, t5 before t3
constraint, t1 starts-at t5
domain, t6 ends-after mon 11am
domain, t3 ends-by fri 3pm 18
domain, t6 ends-by thu 9am 6
domain, t5 ends-by tue 3pm 13
task, t1 3
task, t2 3
task, t3 2
task, t4 1
task, t5 2
task, t6 1
constraint, t6 starts-at t1
constraint, t3 same-day t4
domain, t1 ends-by tue 12pm 16
domain, t5 ends-by tue 12pm 17
domain, t3 ends-by tue 1pm 17
task, t1 2
task, t2 4
task, t3 4
task, t4 4
task, t5 3
task, t6 2
constraint, t4 starts-at t3
constraint, t1 starts-at t6
```

```
domain, t5 ends-after thu 10am
domain, t5 ends-by wed 10am 20
domain, t6 ends-by thu 4pm 14
domain, t3 ends-by fri 2pm 19
domain, t4 ends-by mon 3pm 9
task, t1 4
task, t2 1
task, t3 1
task, t4 4
task, t5 1
task, t6 1
constraint, t6 before t1
constraint, t5 after t3
domain, t2 ends-by fri 11am 8
domain, t3 ends-by fri 4pm 20
domain, t6 ends-by thu 4pm 13
domain, t5 ends-by wed 10am 12
task, t1 2
task, t2 2
task, t3 1
task, t4 4
task, t5 1
task, t6 3
constraint, t5 before t2
domain, t3 ends-before wed 12pm
domain, t6 ends-by tue 11am 16
domain, t4 ends-by fri 4pm 19
domain, t2 ends-by thu 9am 10
domain, t1 ends-by thu 2pm 18
task, t1 4
task, t2 1
task, t3 1
task, t4 3
task, t5 2
task, t6 3
constraint, t2 after t3
constraint, t2 after t4
domain, t3 ends-before fri 11am
domain, t5 ends-by thu 2pm 5
domain, t4 ends-by tue 3pm 20
domain, t3 ends-by fri 12pm 13
domain, t2 ends-by tue 10am 9
task, t1 1
task, t2 3
task, t3 2
task, t4 3
task, t5 4
task, t6 3
constraint, t3 after t2
constraint, t1 starts-at t6
constraint, t3 same-day t4
constraint, t6 before t2
domain, t5 starts-after thu 2pm
domain, t3 ends-by mon 4pm 10
domain, t4 ends-by thu 12pm 20
domain, t5 ends-by fri 12pm 5
domain, t1 ends-by thu 9am 18
task, t1 1
task, t2 1
task, t3 4
```

```
task, t4 1
task, t5 1
task, t6 4
constraint, t3 starts-at t6
constraint, t5 starts-at t3
constraint, t5 after t2
domain, t4 starts-after wed 2pm
domain, t4 ends-by tue 10am 6
domain, t3 ends-by wed 2pm 18
domain, t5 ends-by fri 9am 10
task, t1 4
task, t2 1
task, t3 4
task, t4 3
task, t5 4
task, t6 3
constraint, t3 after t6
constraint, t3 before t1
constraint, t2 same-day t4
domain, t4 ends-by thu 11am 11
domain, t5 ends-by wed 12pm 7
domain, t2 ends-by thu 9am 15
domain, t1 ends-by tue 9am 12
task, t1 1
task, t2 1
task, t3 2
task, t4 1
task, t5 2
task, t6 1
constraint, t5 before t4
constraint, t3 starts-at t5
constraint, t5 starts-at t1
constraint, t6 before t4
domain, t6 ends-before wed 9am
domain, t3 ends-by tue 2pm 15
domain, t2 ends-by wed 10am 13
domain, t4 ends-by wed 11am 5
domain, t6 ends-by tue 3pm 10
task, t1 2
task, t2 2
task, t3 2
task, t4 4
task, t5 2
task, t6 2
constraint, t6 after t3
constraint, t5 after t1
domain, t6 ends-before tue 11am
domain, t3 ends-by thu 12pm 9
domain, t5 ends-by wed 9am 9
domain, t2 ends-by fri 9am 7
domain, t6 ends-by fri 9am 6
task, t1 3
task, t2 3
task, t3 1
task, t4 2
task, t5 4
task, t6 3
constraint, t1 before t3
constraint, t6 after t1
domain, t2 starts-before thu 1pm
```

```
domain, t3 ends-by tue 4pm 9
domain, t1 ends-by wed 10am 8
domain, t4 ends-by fri 2pm 11
task, t1 1
task, t2 4
task, t3 2
task, t4 3
task, t5 3
task, t6 3
constraint, t2 same-day t3
constraint, t4 starts-at t6
constraint, t1 starts-at t3
constraint, t1 starts-at t5
domain, t1 ends-after mon 2pm
domain, t6 ends-by mon 11am 17
domain, t2 ends-by fri 11am 16
domain, t4 ends-by mon 1pm 18
task, t1 3
task, t2 1
task, t3 3
task, t4 2
task, t5 1
task, t6 4
constraint, t6 after t5
constraint, t2 same-day t6
constraint, t3 before t6
domain, t5 ends-by tue 9am 11
domain, t6 ends-by thu 10am 16
domain, t1 ends-by tue 2pm 11
domain, t3 ends-by mon 12pm 13
task, t1 3
task, t2 4
task, t3 3
task, t4 2
task, t5 4
task, t6 4
constraint, t5 starts-at t4
constraint, t5 same-day t2
constraint, t4 starts-at t5
constraint, t1 before t3
domain, t2 ends-by thu 3pm 9
domain, t3 ends-by thu 4pm 7
domain, t1 ends-by thu 2pm 6
domain, t5 ends-by fri 12pm 16
task, t1 3
task, t2 2
task, t3 4
task, t4 2
task, t5 1
task, t6 4
constraint, t6 same-day t3
constraint, t5 same-day t2
domain, t3 ends-before fri 2pm
domain, t4 ends-by thu 1pm 11
domain, t6 ends-by wed 12pm 10
domain, t2 ends-by tue 11am 5
domain, t3 ends-by fri 1pm 13
task, t1 1
task, t2 4
task, t3 2
```

```
task, t4 1
task, t5 1
task, t6 2
constraint, t4 before t2
constraint, t1 same-day t2
domain, t1 ends-by mon 11am 20
domain, t2 ends-by tue 9am 19
domain, t3 ends-by thu 9am 18
domain, t6 ends-by fri 4pm 12
task, t1 1
task, t2 1
task, t3 2
task, t4 4
task, t5 1
task, t6 2
constraint, t5 after t6
constraint, t3 starts-at t6
domain, t3 ends-by tue 3pm 13
domain, t4 ends-by tue 4pm 5
domain, t6 ends-by wed 9am 7
task, t1 3
task, t2 1
task, t3 4
task, t4 2
task, t5 1
task, t6 3
constraint, t5 same-day t1
constraint, t4 same-day t6
constraint, t6 before t4
constraint, t4 same-day t1
domain, t4 starts-before fri 9am
domain, t4 ends-by wed 9am 16
domain, t5 ends-by wed 11am 14
domain, t2 ends-by wed 12pm 9
task, t1 4
task, t2 3
task, t3 2
task, t4 2
task, t5 3
task, t6 2
constraint, t3 starts-at t6
constraint, t2 starts-at t6
domain, t5 ends-by fri 3pm 12
domain, t2 ends-by tue 3pm 6
domain, t6 ends-by wed 4pm 17
domain, t3 ends-by wed 11am 19
task, t1 3
task, t2 4
task, t3 3
task, t4 3
task, t5 4
task, t6 1
constraint, t1 starts-at t4
constraint, t4 same-day t1
constraint, t1 after t3
constraint, t1 starts-at t3
domain, t6 ends-by tue 1pm 15
domain, t3 ends-by wed 10am 19
domain, t5 ends-by fri 10am 16
task, t1 3
```

```
task, t2 1
task, t3 3
task, t4 2
task, t5 3
task, t6 1
constraint, t5 before t2
constraint, t3 same-day t2
constraint, t4 before t6
domain, t1 ends-by thu 3pm 9
domain, t3 ends-by wed 4pm 13
domain, t4 ends-by wed 2pm 5
task, t1 4
task, t2 3
task, t3 2
task, t4 4
task, t5 3
task, t6 2
constraint, t4 after t5
constraint, t5 after t3
constraint, t4 same-day t1
constraint, t1 before t6
domain, t5 starts-after mon 10am
domain, t4 ends-by thu 9am 12
domain, t2 ends-by thu 10am 18
domain, t3 ends-by fri 1pm 10
task, t1 4
task, t2 4
task, t3 1
task, t4 1
task, t5 3
task, t6 1
constraint, t1 starts-at t2
constraint, t5 starts-at t3
constraint, t3 starts-at t6
constraint, t3 same-day t1
domain, t1 ends-before fri 10am
domain, t1 ends-by fri 11am 5
domain, t5 ends-by thu 2pm 13
domain, t3 ends-by fri 1pm 17
domain, t2 ends-by mon 10am 20
task, t1 3
task, t2 3
task, t3 4
task, t4 4
task, t5 2
task, t6 2
constraint, t1 starts-at t6
constraint, t4 before t6
constraint, t6 after t5
domain, t6 ends-before tue 9am
domain, t5 ends-by thu 11am 7
domain, t1 ends-by mon 1pm 15
domain, t4 ends-by fri 3pm 17
domain, t6 ends-by thu 10am 16
task, t1 2
task, t2 2
task, t3 2
task, t4 2
task, t5 2
task, t6 1
```

```
constraint, t6 after t5
constraint, t6 starts-at t2
constraint, t3 starts-at t4
constraint, t1 starts-at t2
domain, t4 ends-by thu 10am 10
domain, t3 ends-by wed 3pm 18
domain, t1 ends-by thu 9am 19
task, t1 2
task, t2 3
task, t3 3
task, t4 4
task, t5 2
task, t6 3
constraint, t1 same-day t6
constraint, t4 after t3
constraint, t1 before t4
constraint, t3 same-day t2
domain, t1 ends-by fri 11am 16
domain, t5 ends-by fri 11am 12
domain, t4 ends-by mon 11am 11
domain, t6 ends-by thu 10am 12
task, t1 3
task, t2 1
task, t3 2
task, t4 3
task, t5 2
task, t6 4
constraint, t4 before t5
constraint, t6 after t3
constraint, t6 starts-at t1
constraint, t4 same-day t3
domain, t2 ends-by mon 4pm 12
domain, t4 ends-by mon 9am 18
domain, t3 ends-by mon 9am 6
domain, t5 ends-by fri 2pm 8
task, t1 2
task, t2 1
task, t3 4
task, t4 3
task, t5 2
task, t6 3
constraint, t2 before t6
constraint, t4 starts-at t6
constraint, t4 after t1
domain, t6 ends-by fri 12pm 5
domain, t1 ends-by wed 10am 10
domain, t5 ends-by wed 9am 20
domain, t2 ends-by tue 12pm 5
task, t1 1
task, t2 4
task, t3 1
task, t4 2
task, t5 1
task, t6 2
constraint, t4 same-day t1
constraint, t3 same-day t4
constraint, t2 same-day t1
constraint, t5 same-day t3
domain, t3 ends-after fri 3pm
domain, t1 ends-by mon 1pm 20
```

```
domain, t3 ends-by thu 11am 13
domain, t5 ends-by wed 9am 15
task, t1 1
task, t2 2
task, t3 3
task, t4 3
task, t5 4
task, t6 4
constraint, t3 before t6
constraint, t6 starts-at t3
constraint, t3 starts-at t5
domain, t6 ends-by tue 2pm 10
domain, t4 ends-by fri 4pm 5
domain, t3 ends-by tue 10am 14
domain, t5 ends-by mon 2pm 7
task, t1 4
task, t2 4
task, t3 2
task, t4 3
task, t5 2
task, t6 4
constraint, t4 after t6
constraint, t2 same-day t3
constraint, t6 before t2
constraint, t2 after t1
domain, t1 starts-before tue 11am
domain, t6 ends-by fri 2pm 15
domain, t2 ends-by wed 2pm 17
domain, t5 ends-by wed 3pm 20
domain, t1 ends-by wed 2pm 14
task, t1 3
task, t2 3
task, t3 3
task, t4 3
task, t5 2
task, t6 4
task, t7 2
task, t8 3
constraint, t2 before t5
constraint, t8 starts-at t2
constraint, t4 after t3
constraint, t5 starts-at t3
constraint, t1 after t4
domain, t4 starts-after mon 12pm
domain, t4 ends-by wed 12pm 6
domain, t8 ends-by mon 4pm 6
domain, t1 ends-by wed 12pm 16
domain, t6 ends-by mon 9am 5
domain, t5 ends-by mon 12pm 5
task, t1 4
task, t2 3
task, t3 3
task, t4 4
task, t5 4
task, t6 4
task, t7 3
task, t8 2
constraint, t5 before t7
constraint, t5 starts-at t6
constraint, t7 starts-at t1
```

```
domain, t4 ends-by tue 2pm 16
domain, t7 ends-by fri 3pm 17
domain, t6 ends-by mon 9am 6
domain, t3 ends-by mon 12pm 12
domain, t5 ends-by wed 1pm 15
task, t1 1
task, t2 2
task, t3 3
task, t4 3
task, t5 3
task, t6 2
task, t7 2
task, t8 2
constraint, t3 same-day t4
constraint, t8 before t1
constraint, t4 same-day t2
constraint, t6 starts-at t4
constraint, t3 before t2
domain, t3 ends-by thu 11am 20
domain, t4 ends-by mon 12pm 12
domain, t1 ends-by thu 4pm 9
domain, t5 ends-by mon 12pm 19
task, t1 2
task, t2 1
task, t3 4
task, t4 4
task, t5 3
task, t6 1
task, t7 3
task, t8 2
constraint, t6 starts-at t2
constraint, t5 same-day t3
constraint, t4 starts-at t2
domain, t2 ends-after mon 1pm
domain, t4 ends-by wed 12pm 8
domain, t6 ends-by tue 10am 12
domain, t1 ends-by tue 3pm 20
domain, t2 ends-by wed 4pm 13
task, t1 3
task, t2 4
task, t3 4
task, t4 3
task, t5 3
task, t6 2
task, t7 4
task, t8 1
constraint, t5 before t1
constraint, t3 starts-at t5
domain, t4 ends-before mon 3pm
domain, t3 ends-by thu 4pm 12
domain, t1 ends-by mon 9am 11
domain, t8 ends-by mon 2pm 6
domain, t2 ends-by wed 9am 16
domain, t6 ends-by tue 4pm 7
task, t1 2
task, t2 1
task, t3 1
task, t4 1
task, t5 4
task, t6 1
```

```
task, t7 1
task, t8 2
constraint, t1 starts-at t4
constraint, t4 starts-at t7
constraint, t7 before t2
constraint, t3 same-day t4
domain, t2 ends-by thu 9am 18
domain, t6 ends-by wed 12pm 10
domain, t4 ends-by wed 4pm 7
domain, t5 ends-by tue 10am 16
domain, t7 ends-by mon 4pm 6
task, t1 1
task, t2 1
task, t3 2
task, t4 3
task, t5 1
task, t6 1
task, t7 4
task, t8 3
constraint, t4 same-day t8
constraint, t7 before t2
constraint, t6 before t4
constraint, t4 starts-at t6
constraint, t6 same-day t2
domain, t1 ends-by thu 12pm 16
domain, t2 ends-by tue 11am 6
domain, t5 ends-by tue 12pm 9
domain, t8 ends-by mon 9am 20
task, t1 2
task, t2 1
task, t3 2
task, t4 2
task, t5 3
task, t6 3
task, t7 2
task, t8 1
constraint, t6 starts-at t1
constraint, t6 same-day t5
constraint, t6 before t2
domain, t6 ends-after fri 10am
domain, t2 starts-after tue 2pm
domain, t7 ends-by mon 2pm 12
domain, t8 ends-by wed 12pm 13
domain, t4 ends-by thu 10am 19
domain, t3 ends-by wed 3pm 10
domain, t2 ends-by tue 2pm 20
domain, t1 ends-by thu 12pm 17
task, t1 4
task, t2 4
task, t3 1
task, t4 3
task, t5 2
task, t6 2
task, t7 4
task, t8 3
constraint, t5 same-day t4
constraint, t4 after t2
domain, t5 ends-by thu 2pm 19
domain, t3 ends-by wed 2pm 12
domain, t6 ends-by mon 1pm 8
```

```
domain, t1 ends-by wed 1pm 12
task, t1 2
task, t2 1
task, t3 1
task, t4 3
task, t5 3
task, t6 2
task, t7 3
task, t8 3
constraint, t1 before t7
constraint, t7 before t5
domain, t1 ends-by thu 10am 8
domain, t7 ends-by tue 11am 5
domain, t8 ends-by fri 11am 15
domain, t3 ends-by thu 4pm 6
task, t1 1
task, t2 1
task, t3 4
task, t4 4
task, t5 3
task, t6 1
task, t7 4
task, t8 3
constraint, t3 before t2
constraint, t5 before t3
domain, t1 ends-before thu 2pm
domain, t5 ends-after fri 10am
domain, t6 ends-by thu 2pm 6
domain, t8 ends-by mon 1pm 8
domain, t3 ends-by wed 2pm 7
domain, t7 ends-by wed 12pm 14
domain, t1 ends-by mon 9am 10
domain, t2 ends-by fri 1pm 10
task, t1 1
task, t2 2
task, t3 2
task, t4 3
task, t5 4
task, t6 2
task, t7 2
task, t8 1
constraint, t2 same-day t5
constraint, t7 same-day t5
constraint, t6 same-day t2
constraint, t8 before t1
constraint, t1 before t6
domain, t6 ends-after thu 3pm
domain, t8 ends-by wed 3pm 8
domain, t2 ends-by wed 11am 10
domain, t7 ends-by wed 9am 16
domain, t1 ends-by thu 12pm 13
domain, t4 ends-by mon 4pm 19
domain, t3 ends-by tue 9am 13
task, t1 4
task, t2 3
task, t3 2
task, t4 2
task, t5 3
task, t6 4
task, t7 2
```

```
task, t8 3
constraint, t2 before t7
constraint, t2 after t5
domain, t4 starts-before tue 12pm
domain, t7 ends-by mon 2pm 16
domain, t5 ends-by thu 1pm 7
domain, t2 ends-by fri 2pm 8
domain, t3 ends-by mon 11am 16
domain, t4 ends-by wed 12pm 8
domain, t1 ends-by tue 3pm 13
task, t1 4
task, t2 4
task, t3 3
task, t4 2
task, t5 3
task, t6 1
task, t7 2
task, t8 4
constraint, t5 before t6
constraint, t6 starts-at t2
constraint, t3 after t7
domain, t6 ends-after mon 11am
domain, t4 ends-by mon 1pm 16
domain, t2 ends-by wed 9am 15
domain, t5 ends-by thu 1pm 11
domain, t8 ends-by tue 2pm 11
domain, t3 ends-by tue 2pm 5
task, t1 3
task, t2 1
task, t3 3
task, t4 3
task, t5 2
task, t6 2
task, t7 4
task, t8 1
constraint, t6 after t2
constraint, t2 same-day t6
constraint, t4 starts-at t1
domain, t8 ends-before mon 11am
domain, t5 ends-by wed 10am 17
domain, t3 ends-by thu 2pm 19
domain, t1 ends-by tue 4pm 10
domain, t7 ends-by mon 4pm 12
task, t1 2
task, t2 3
task, t3 2
task, t4 1
task, t5 2
task, t6 3
task, t7 4
task, t8 1
constraint, t1 after t2
constraint, t5 after t3
constraint, t3 starts-at t4
constraint, t5 same-day t1
constraint, t2 starts-at t5
domain, t4 starts-after tue 3pm
domain, t7 starts-after tue 12pm
domain, t3 ends-by thu 3pm 8
domain, t6 ends-by thu 11am 11
```

```
domain, t8 ends-by mon 1pm 7
domain, t7 ends-by thu 10am 13
domain, t4 ends-by tue 4pm 14
domain, t1 ends-by mon 9am 6
task, t1 1
task, t2 4
task, t3 4
task, t4 4
task, t5 3
task, t6 3
task, t7 1
task, t8 3
constraint, t7 after t1
constraint, t1 before t5
domain, t8 ends-before fri 1pm
domain, t4 ends-by fri 11am 19
domain, t5 ends-by tue 12pm 11
domain, t2 ends-by fri 1pm 20
domain, t3 ends-by wed 3pm 13
domain, t7 ends-by thu 4pm 17
domain, t1 ends-by thu 3pm 10
task, t1 2
task, t2 2
task, t3 1
task, t4 3
task, t5 4
task, t6 3
task, t7 3
task, t8 3
constraint, t5 after t3
constraint, t3 same-day t5
constraint, t5 after t2
domain, t8 ends-by thu 12pm 9
domain, t3 ends-by fri 3pm 11
domain, t1 ends-by fri 2pm 15
domain, t6 ends-by fri 9am 5
domain, t2 ends-by thu 9am 16
domain, t7 ends-by tue 11am 19
task, t1 3
task, t2 4
task, t3 4
task, t4 2
task, t5 4
task, t6 4
task, t7 2
task, t8 3
constraint, t5 before t8
constraint, t6 starts-at t7
constraint, t6 starts-at t3
constraint, t1 after t3
domain, t2 ends-by wed 3pm 16
domain, t6 ends-by mon 12pm 9
domain, t8 ends-by mon 4pm 9
domain, t7 ends-by mon 11am 11
domain, t4 ends-by fri 4pm 19
task, t1 3
task, t2 3
task, t3 2
task, t4 4
task, t5 1
```

```
task, t6 3
task, t7 3
task, t8 2
constraint, t8 after t6
constraint, t8 same-day t3
constraint, t5 same-day t2
constraint, t2 after t7
constraint, t8 before t7
domain, t6 ends-by thu 1pm 11
domain, t8 ends-by wed 3pm 14
domain, t2 ends-by thu 9am 9
domain, t7 ends-by tue 11am 16
domain, t4 ends-by tue 10am 17
domain, t3 ends-by mon 2pm 11
task, t1 4
task, t2 2
task, t3 4
task, t4 3
task, t5 1
task, t6 3
task, t7 4
task, t8 3
constraint, t4 after t7
constraint, t2 same-day t7
constraint, t4 before t3
domain, t7 ends-before thu 1pm
domain, t7 ends-by thu 11am 9
domain, t5 ends-by fri 3pm 10
domain, t8 ends-by fri 9am 5
domain, t2 ends-by mon 1pm 13
domain, t3 ends-by tue 3pm 19
domain, t4 ends-by mon 12pm 18
task, t1 1
task, t2 2
task, t3 3
task, t4 4
task, t5 3
task, t6 3
task, t7 1
task, t8 4
constraint, t7 same-day t4
constraint, t4 after t1
constraint, t2 after t7
constraint, t2 after t6
constraint, t7 after t3
domain, t4 ends-after wed 1pm
domain, t7 ends-before mon 4pm
domain, t6 ends-by fri 12pm 6
domain, t7 ends-by wed 9am 6
domain, t3 ends-by fri 12pm 9
domain, t4 ends-by mon 1pm 17
domain, t5 ends-by mon 10am 6
domain, t2 ends-by tue 4pm 13
task, t1 1
task, t2 2
task, t3 1
task, t4 2
task, t5 1
task, t6 4
task, t7 3
```

```
task, t8 3
constraint, t1 starts-at t3
constraint, t5 after t8
domain, t2 ends-after fri 4pm
domain, t1 starts-before mon 4pm
domain, t8 ends-by tue 1pm 18
domain, t1 ends-by wed 4pm 14
domain, t3 ends-by mon 10am 6
domain, t4 ends-by mon 12pm 20
domain, t5 ends-by fri 10am 11
domain, t6 ends-by fri 3pm 14
task, t1 3
task, t2 3
task, t3 4
task, t4 2
task, t5 3
task, t6 2
task, t7 3
task, t8 2
constraint, t2 before t3
constraint, t3 before t2
constraint, t1 starts-at t6
domain, t3 ends-by tue 4pm 16
domain, t4 ends-by wed 9am 18
domain, t5 ends-by mon 3pm 18
domain, t2 ends-by fri 10am 5
domain, t1 ends-by wed 11am 18
task, t1 4
task, t2 1
task, t3 1
task, t4 4
task, t5 2
task, t6 3
task, t7 3
task, t8 1
constraint, t5 after t7
constraint, t6 after t1
domain, t1 starts-after tue 4pm
domain, t4 ends-by fri 11am 15
domain, t1 ends-by mon 2pm 18
domain, t8 ends-by thu 2pm 9
domain, t3 ends-by fri 1pm 8
domain, t7 ends-by wed 11am 11
task, t1 1
task, t2 4
task, t3 2
task, t4 3
task, t5 2
task, t6 3
task, t7 3
task, t8 4
constraint, t6 starts-at t5
constraint, t4 starts-at t5
constraint, t8 before t1
constraint, t5 same-day t1
domain, t1 ends-by fri 2pm 8
domain, t8 ends-by thu 2pm 11
domain, t2 ends-by thu 10am 13
domain, t5 ends-by wed 9am 9
domain, t7 ends-by fri 2pm 5
```

```
task, t1 1
task, t2 1
task, t3 4
task, t4 3
task, t5 3
task, t6 1
task, t7 2
task, t8 4
constraint, t7 before t5
constraint, t4 same-day t6
constraint, t8 starts-at t6
constraint, t6 after t4
constraint, t2 starts-at t5
domain, t3 starts-after fri 9am
domain, t5 starts-before fri 3pm
domain, t3 ends-by mon 11am 9
domain, t4 ends-by wed 3pm 9
domain, t6 ends-by mon 3pm 14
domain, t8 ends-by mon 11am 11
domain, t5 ends-by fri 2pm 6
domain, t2 ends-by wed 3pm 6
task, t1 3
task, t2 2
task, t3 3
task, t4 2
task, t5 4
task, t6 3
task, t7 4
task, t8 4
constraint, t7 before t6
constraint, t7 after t4
constraint, t6 before t7
domain, t8 ends-by wed 11am 19
domain, t2 ends-by fri 9am 11
domain, t7 ends-by mon 3pm 15
domain, t4 ends-by fri 12pm 18
task, t1 3
task, t2 2
task, t3 3
task, t4 2
task, t5 3
task, t6 4
task, t7 2
task, t8 4
constraint, t3 before t7
constraint, t4 before t6
constraint, t7 before t8
constraint, t8 after t1
domain, t5 ends-before fri 2pm
domain, t4 starts-after wed 4pm
domain, t2 ends-by mon 4pm 17
domain, t1 ends-by tue 10am 7
domain, t7 ends-by mon 11am 15
domain, t8 ends-by wed 9am 8
domain, t5 ends-by mon 11am 16
task, t1 2
task, t2 2
task, t3 3
task, t4 1
task, t5 2
```

```
task, t6 2
task, t7 3
task, t8 4
constraint, t1 before t7
constraint, t4 after t7
constraint, t6 same-day t1
domain, t2 ends-after wed 9am
domain, t3 ends-by wed 1pm 16
domain, t8 ends-by tue 10am 6
domain, t1 ends-by thu 10am 15
domain, t7 ends-by fri 9am 7
task, t1 2
task, t2 4
task, t3 1
task, t4 1
task, t5 4
task, t6 3
task, t7 2
task, t8 4
constraint, t6 starts-at t3
constraint, t2 starts-at t8
constraint, t7 starts-at t8
constraint, t7 before t1
domain, t4 ends-by tue 12pm 14
domain, t2 ends-by wed 3pm 17
domain, t1 ends-by fri 4pm 19
domain, t8 ends-by fri 4pm 15
task, t1 3
task, t2 4
task, t3 2
task, t4 4
task, t5 3
task, t6 1
task, t7 3
task, t8 2
constraint, t6 starts-at t2
constraint, t4 starts-at t6
constraint, t6 after t2
constraint, t5 starts-at t8
domain, t3 starts-before thu 1pm
domain, t2 starts-before wed 3pm
domain, t4 ends-by thu 9am 12
domain, t8 ends-by wed 11am 17
domain, t3 ends-by fri 10am 6
domain, t6 ends-by wed 4pm 20
domain, t1 ends-by fri 1pm 8
task, t1 3
task, t2 4
task, t3 3
task, t4 3
task, t5 3
task, t6 2
task, t7 3
task, t8 4
constraint, t4 starts-at t3
constraint, t2 after t4
domain, t5 ends-by fri 9am 13
domain, t7 ends-by fri 2pm 17
domain, t3 ends-by wed 10am 8
domain, t1 ends-by mon 1pm 20
```

```
task, t1 1
task, t2 1
task, t3 1
task, t4 4
task, t5 4
task, t6 4
task, t7 2
task, t8 4
constraint, t2 before t7
constraint, t7 before t3
constraint, t7 before t6
domain, t4 ends-after thu 2pm
domain, t8 ends-by wed 2pm 12
domain, t3 ends-by fri 3pm 18
domain, t7 ends-by mon 1pm 20
domain, t6 ends-by fri 4pm 18
domain, t4 ends-by fri 11am 16
domain, t2 ends-by wed 2pm 17
task, t1 1
task, t2 2
task, t3 4
task, t4 4
task, t5 4
task, t6 4
task, t7 2
task, t8 2
constraint, t4 before t2
constraint, t7 before t8
constraint, t3 after t4
constraint, t4 before t6
domain, t4 ends-by fri 10am 8
domain, t1 ends-by tue 9am 8
domain, t2 ends-by wed 2pm 14
domain, t3 ends-by wed 9am 18
domain, t5 ends-by mon 4pm 9
task, t1 3
task, t2 4
task, t3 4
task, t4 3
task, t5 4
task, t6 2
task, t7 3
task, t8 2
constraint, t6 before t7
constraint, t5 before t1
constraint, t7 starts-at t2
domain, t7 starts-after mon 3pm
domain, t5 starts-after mon 3pm
domain, t7 ends-by wed 11am 11
domain, t4 ends-by thu 12pm 14
domain, t1 ends-by tue 10am 12
domain, t8 ends-by fri 12pm 16
task, t1 2
task, t2 1
task, t3 2
task, t4 2
task, t5 3
task, t6 4
task, t7 1
task, t8 4
```

```
constraint, t6 starts-at t3
constraint, t8 starts-at t4
constraint, t7 after t8
constraint, t6 before t1
domain, t7 ends-by fri 1pm 12
domain, t3 ends-by fri 1pm 17
domain, t6 ends-by tue 9am 7
domain, t1 ends-by mon 1pm 16
domain, t8 ends-by tue 3pm 11
task, t1 3
task, t2 3
task, t3 2
task, t4 2
task, t5 3
task, t6 4
task, t7 1
task, t8 3
constraint, t3 starts-at t2
constraint, t4 before t7
domain, t3 ends-by mon 10am 7
domain, t6 ends-by thu 4pm 5
domain, t5 ends-by mon 4pm 15
domain, t7 ends-by thu 11am 13
task, t1 2
task, t2 3
task, t3 4
task, t4 1
task, t5 1
task, t6 4
task, t7 2
task, t8 3
constraint, t5 before t7
constraint, t6 after t8
constraint, t1 starts-at t4
constraint, t4 starts-at t1
domain, t7 ends-by fri 2pm 11
domain, t1 ends-by thu 3pm 13
domain, t4 ends-by fri 1pm 7
domain, t2 ends-by wed 4pm 19
domain, t6 ends-by wed 2pm 7
domain, t8 ends-by fri 11am 17
task, t1 1
task, t2 2
task, t3 1
task, t4 3
task, t5 2
task, t6 3
task, t7 3
task, t8 3
constraint, t3 before t7
constraint, t6 starts-at t1
domain, t2 starts-after thu 3pm
domain, t3 ends-before thu 9am
domain, t4 ends-by tue 12pm 8
domain, t8 ends-by tue 3pm 15
domain, t3 ends-by thu 10am 5
domain, t5 ends-by thu 2pm 11
domain, t2 ends-by tue 10am 18
domain, t6 ends-by wed 2pm 17
task, t1 3
```

```
task, t2 2
task, t3 4
task, t4 3
task, t5 1
task, t6 2
task, t7 4
task, t8 2
task, t9 2
task, t10 2
constraint, t5 same-day t2
constraint, t10 same-day t1
constraint, t1 same-day t9
domain, t10 ends-by wed 9am 17
domain, t2 ends-by tue 4pm 8
domain, t4 ends-by thu 4pm 6
domain, t6 ends-by mon 4pm 7
domain, t9 ends-by wed 4pm 17
domain, t5 ends-by wed 3pm 8
domain, t7 ends-by tue 2pm 18
task, t1 2
task, t2 2
task, t3 2
task, t4 4
task, t5 1
task, t6 2
task, t7 2
task, t8 2
task, t9 3
task, t10 3
constraint, t9 before t6
constraint, t3 same-day t7
constraint, t6 same-day t1
constraint, t8 after t9
constraint, t3 starts-at t8
constraint, t7 starts-at t1
domain, t7 starts-after fri 3pm
domain, t8 ends-after fri 4pm
domain, t8 ends-by mon 1pm 10
domain, t9 ends-by wed 9am 20
domain, t3 ends-by mon 9am 12
domain, t5 ends-by wed 9am 13
domain, t7 ends-by thu 10am 9
domain, t6 ends-by thu 3pm 13
domain, t10 ends-by thu 2pm 12
task, t1 2
task, t2 2
task, t3 3
task, t4 2
task, t5 2
task, t6 2
task, t7 2
task, t8 3
task, t9 3
task, t10 4
constraint, t4 same-day t1
constraint, t6 starts-at t2
constraint, t4 after t2
constraint, t7 before t6
constraint, t4 same-day t7
constraint, t8 after t10
```

```
domain, t2 ends-by fri 2pm 18
domain, t3 ends-by fri 9am 20
domain, t6 ends-by wed 10am 19
domain, t8 ends-by fri 9am 15
domain, t10 ends-by wed 3pm 8
domain, t9 ends-by mon 10am 9
domain, t5 ends-by thu 1pm 14
domain, t1 ends-by tue 10am 9
task, t1 4
task, t2 2
task, t3 4
task, t4 2
task, t5 4
task, t6 1
task, t7 1
task, t8 3
task, t9 4
task, t10 4
constraint, t10 same-day t1
constraint, t8 starts-at t1
constraint, t10 after t1
constraint, t1 starts-at t9
domain, t7 starts-after thu 3pm
domain, t10 ends-before tue 4pm
domain, t6 ends-by wed 3pm 5
domain, t4 ends-by fri 12pm 16
domain, t10 ends-by fri 10am 20
domain, t7 ends-by thu 10am 15
domain, t2 ends-by fri 10am 13
domain, t9 ends-by tue 4pm 20
domain, t3 ends-by mon 1pm 9
task, t1 1
task, t2 4
task, t3 1
task, t4 3
task, t5 3
task, t6 4
task, t7 1
task, t8 1
task, t9 1
task, t10 4
constraint, t10 before t8
constraint, t6 starts-at t7
constraint, t7 same-day t2
domain, t8 ends-by mon 2pm 10
domain, t6 ends-by thu 9am 6
domain, t4 ends-by tue 3pm 14
domain, t7 ends-by tue 4pm 17
domain, t3 ends-by fri 3pm 19
domain, t5 ends-by tue 4pm 13
task, t1 1
task, t2 4
task, t3 3
task, t4 2
task, t5 1
task, t6 1
task, t7 4
task, t8 2
task, t9 4
task, t10 1
```

```
constraint, t3 starts-at t10
constraint, t4 starts-at t7
constraint, t4 before t9
domain, t9 ends-before wed 9am
domain, t6 ends-before wed 3pm
domain, t10 ends-by fri 1pm 9
domain, t2 ends-by tue 12pm 7
domain, t4 ends-by fri 1pm 16
domain, t7 ends-by tue 4pm 16
domain, t3 ends-by thu 12pm 17
domain, t8 ends-by wed 12pm 12
task, t1 2
task, t2 4
task, t3 3
task, t4 1
task, t5 2
task, t6 1
task, t7 2
task, t8 2
task, t9 4
task, t10 1
constraint, t5 after t7
constraint, t3 before t7
constraint, t4 before t3
domain, t1 starts-after wed 3pm
domain, t3 ends-before thu 3pm
domain, t7 ends-by thu 4pm 16
domain, t4 ends-by wed 12pm 11
domain, t1 ends-by tue 1pm 17
domain, t9 ends-by tue 3pm 8
domain, t6 ends-by thu 9am 14
domain, t3 ends-by tue 9am 9
domain, t8 ends-by wed 4pm 18
domain, t2 ends-by thu 12pm 16
task, t1 4
task, t2 2
task, t3 2
task, t4 2
task, t5 3
task, t6 4
task, t7 2
task, t8 2
task, t9 3
task, t10 3
constraint, t10 same-day t3
constraint, t6 after t10
constraint, t8 same-day t6
domain, t8 ends-by thu 9am 15
domain, t5 ends-by wed 9am 18
domain, t1 ends-by tue 9am 10
domain, t3 ends-by thu 11am 19
domain, t2 ends-by tue 4pm 9
task, t1 3
task, t2 3
task, t3 2
task, t4 3
task, t5 1
task, t6 1
task, t7 4
task, t8 1
```

```
task, t9 3
task, t10 3
constraint, t9 after t2
constraint, t9 same-day t10
constraint, t1 starts-at t3
domain, t10 ends-by mon 2pm 19
domain, t5 ends-by mon 4pm 11
domain, t2 ends-by fri 10am 15
domain, t8 ends-by thu 9am 13
domain, t6 ends-by fri 3pm 9
domain, t4 ends-by mon 4pm 10
domain, t3 ends-by fri 4pm 19
domain, t1 ends-by wed 10am 19
task, t1 3
task, t2 1
task, t3 4
task, t4 4
task, t5 1
task, t6 4
task, t7 2
task, t8 3
task, t9 1
task, t10 2
constraint, t5 starts-at t4
constraint, t5 starts-at t6
constraint, t9 same-day t10
constraint, t6 after t7
domain, t4 ends-by fri 3pm 6
domain, t2 ends-by thu 3pm 12
domain, t1 ends-by wed 10am 5
domain, t3 ends-by thu 10am 10
domain, t9 ends-by tue 9am 13
domain, t5 ends-by tue 4pm 7
domain, t7 ends-by wed 12pm 15
task, t1 3
task, t2 2
task, t3 1
task, t4 1
task, t5 3
task, t6 3
task, t7 4
task, t8 2
task, t9 1
task, t10 2
constraint, t4 after t8
constraint, t7 after t5
constraint, t9 before t1
constraint, t1 starts-at t7
domain, t3 starts-before thu 10am
domain, t3 ends-by tue 3pm 10
domain, t2 ends-by fri 2pm 7
domain, t10 ends-by fri 11am 14
domain, t8 ends-by tue 1pm 19
domain, t1 ends-by wed 4pm 17
task, t1 3
task, t2 3
task, t3 3
task, t4 1
task, t5 1
task, t6 2
```

```
task, t7 3
task, t8 2
task, t9 4
task, t10 4
constraint, t10 before t6
constraint, t1 before t7
constraint, t1 starts-at t9
constraint, t1 before t3
constraint, t5 after t1
domain, t4 ends-before wed 10am
domain, t10 ends-before tue 9am
domain, t1 ends-by wed 12pm 13
domain, t6 ends-by wed 11am 12
domain, t9 ends-by mon 2pm 17
domain, t8 ends-by wed 11am 18
domain, t7 ends-by mon 1pm 12
domain, t3 ends-by fri 4pm 10
domain, t2 ends-by thu 1pm 7
domain, t4 ends-by fri 11am 9
task, t1 3
task, t2 2
task, t3 4
task, t4 3
task, t5 1
task, t6 2
task, t7 3
task, t8 1
task, t9 2
task, t10 2
constraint, t10 before t3
constraint, t1 same-day t5
constraint, t7 before t2
constraint, t1 before t5
domain, t1 ends-after thu 12pm
domain, t6 ends-before tue 4pm
domain, t7 ends-by tue 2pm 13
domain, t6 ends-by wed 4pm 13
domain, t1 ends-by tue 3pm 18
domain, t4 ends-by wed 4pm 18
domain, t10 ends-by mon 2pm 18
domain, t8 ends-by wed 11am 13
task, t1 2
task, t2 3
task, t3 1
task, t4 4
task, t5 4
task, t6 4
task, t7 2
task, t8 2
task, t9 2
task, t10 1
constraint, t6 before t2
constraint, t4 before t5
constraint, t5 after t8
domain, t8 ends-before fri 9am
domain, t2 ends-by tue 12pm 15
domain, t8 ends-by tue 1pm 8
domain, t1 ends-by wed 11am 7
domain, t3 ends-by wed 4pm 18
domain, t10 ends-by wed 11am 16
```

```
domain, t7 ends-by mon 11am 19
domain, t4 ends-by wed 10am 17
domain, t5 ends-by tue 1pm 15
task, t1 3
task, t2 2
task, t3 3
task, t4 2
task, t5 4
task, t6 3
task, t7 1
task, t8 1
task, t9 1
task, t10 2
constraint, t10 after t3
constraint, t5 before t10
constraint, t2 same-day t7
constraint, t3 same-day t8
domain, t4 ends-by tue 3pm 13
domain, t5 ends-by mon 12pm 5
domain, t3 ends-by tue 2pm 16
domain, t9 ends-by fri 10am 10
domain, t1 ends-by thu 3pm 20
domain, t8 ends-by wed 2pm 6
domain, t6 ends-by tue 11am 13
task, t1 2
task, t2 4
task, t3 1
task, t4 4
task, t5 3
task, t6 4
task, t7 1
task, t8 4
task, t9 4
task, t10 1
constraint, t8 after t7
constraint, t10 before t3
constraint, t8 before t3
constraint, t8 after t3
constraint, t2 starts-at t10
domain, t8 starts-before tue 3pm
domain, t9 ends-by mon 3pm 11
domain, t1 ends-by wed 12pm 7
domain, t2 ends-by mon 4pm 19
domain, t10 ends-by mon 10am 7
domain, t4 ends-by thu 2pm 16
task, t1 2
task, t2 3
task, t3 1
task, t4 4
task, t5 1
task, t6 2
task, t7 2
task, t8 4
task, t9 3
task, t10 2
constraint, t7 starts-at t2
constraint, t9 after t1
constraint, t10 before t6
constraint, t5 starts-at t10
constraint, t4 before t9
```

```
domain, t7 ends-by wed 1pm 17
domain, t2 ends-by mon 4pm 18
domain, t6 ends-by wed 11am 15
domain, t1 ends-by wed 9am 18
domain, t9 ends-by wed 4pm 10
task, t1 4
task, t2 3
task, t3 3
task, t4 3
task, t5 3
task, t6 2
task, t7 3
task, t8 2
task, t9 3
task, t10 3
constraint, t5 after t9
constraint, t4 before t8
constraint, t4 before t10
constraint, t10 same-day t1
constraint, t2 same-day t1
constraint, t9 starts-at t2
domain, t1 ends-by tue 3pm 19
domain, t3 ends-by fri 11am 11
domain, t2 ends-by wed 3pm 8
domain, t6 ends-by fri 2pm 18
domain, t8 ends-by thu 2pm 15
domain, t7 ends-by thu 10am 19
domain, t9 ends-by fri 9am 19
task, t1 4
task, t2 3
task, t3 4
task, t4 3
task, t5 3
task, t6 3
task, t7 4
task, t8 1
task, t9 4
task, t10 4
constraint, t4 before t7
constraint, t2 same-day t6
constraint, t7 same-day t6
domain, t6 ends-by mon 12pm 14
domain, t1 ends-by thu 3pm 7
domain, t9 ends-by fri 4pm 17
domain, t4 ends-by wed 4pm 19
domain, t5 ends-by fri 3pm 8
domain, t2 ends-by tue 1pm 8
task, t1 3
task, t2 3
task, t3 3
task, t4 4
task, t5 3
task, t6 4
task, t7 1
task, t8 2
task, t9 4
task, t10 2
constraint, t8 same-day t4
constraint, t3 after t7
constraint, t5 before t9
```

```
constraint, t7 starts-at t10
domain, t1 starts-before tue 11am
domain, t10 ends-after wed 3pm
domain, t9 ends-by thu 9am 7
domain, t2 ends-by tue 3pm 8
domain, t7 ends-by wed 9am 10
domain, t3 ends-by thu 10am 6
domain, t10 ends-by mon 10am 11
task, t1 3
task, t2 4
task, t3 3
task, t4 1
task, t5 1
task, t6 4
task, t7 2
task, t8 3
task, t9 2
task, t10 4
constraint, t6 starts-at t3
constraint, t10 same-day t5
constraint, t2 starts-at t4
constraint, t7 after t10
constraint, t5 same-day t8
constraint, t8 before t5
domain, t7 starts-before fri 9am
domain, t10 ends-after thu 10am
domain, t4 ends-by tue 10am 10
domain, t1 ends-by tue 10am 10
domain, t7 ends-by wed 1pm 5
domain, t9 ends-by thu 12pm 7
domain, t2 ends-by wed 1pm 16
domain, t6 ends-by wed 11am 18
task, t1 3
task, t2 4
task, t3 1
task, t4 1
task, t5 3
task, t6 4
task, t7 2
task, t8 1
task, t9 1
task, t10 1
constraint, t6 starts-at t2
constraint, t5 same-day t4
constraint, t7 same-day t2
constraint, t5 before t7
constraint, t9 after t7
domain, t8 ends-by tue 3pm 8
domain, t2 ends-by mon 9am 19
domain, t9 ends-by tue 4pm 12
domain, t5 ends-by wed 1pm 17
domain, t7 ends-by wed 10am 12
domain, t3 ends-by fri 2pm 7
domain, t6 ends-by tue 3pm 6
task, t1 3
task, t2 2
task, t3 4
task, t4 1
task, t5 3
task, t6 1
```

```
task, t7 2
task, t8 1
task, t9 4
task, t10 4
constraint, t5 after t1
constraint, t10 same-day t6
constraint, t5 after t7
constraint, t8 same-day t3
domain, t9 ends-by mon 1pm 17
domain, t1 ends-by mon 3pm 19
domain, t10 ends-by tue 3pm 11
domain, t8 ends-by tue 11am 17
domain, t7 ends-by tue 11am 17
domain, t6 ends-by tue 1pm 8
domain, t2 ends-by thu 1pm 5
task, t1 4
task, t2 1
task, t3 4
task, t4 1
task, t5 3
task, t6 2
task, t7 1
task, t8 2
task, t9 2
task, t10 3
constraint, t2 starts-at t4
constraint, t3 after t4
constraint, t5 before t6
constraint, t3 after t10
domain, t6 ends-by thu 2pm 19
domain, t10 ends-by wed 10am 5
domain, t9 ends-by mon 11am 12
domain, t2 ends-by thu 12pm 10
domain, t8 ends-by wed 3pm 13
domain, t7 ends-by tue 10am 15
domain, t3 ends-by thu 9am 8
task, t1 3
task, t2 3
task, t3 1
task, t4 1
task, t5 4
task, t6 2
task, t7 4
task, t8 4
task, t9 4
task, t10 2
constraint, t3 before t4
constraint, t1 after t8
constraint, t10 starts-at t4
constraint, t4 after t9
constraint, t4 same-day t2
constraint, t5 after t4
domain, t7 ends-before wed 12pm
domain, t7 ends-by mon 1pm 12
domain, t3 ends-by fri 4pm 7
domain, t2 ends-by mon 2pm 17
domain, t1 ends-by fri 10am 9
domain, t4 ends-by thu 12pm 19
task, t1 4
task, t2 3
```

```
task, t3 3
task, t4 3
task, t5 4
task, t6 1
task, t7 3
task, t8 3
task, t9 4
task, t10 1
constraint, t9 starts-at t1
constraint, t8 same-day t2
constraint, t1 starts-at t4
constraint, t9 starts-at t4
constraint, t1 starts-at t8
constraint, t5 before t7
domain, t4 ends-before tue 4pm
domain, t1 ends-by wed 1pm 19
domain, t2 ends-by mon 12pm 14
domain, t8 ends-by mon 9am 16
domain, t6 ends-by wed 4pm 7
domain, t9 ends-by tue 11am 18
domain, t3 ends-by wed 11am 19
domain, t5 ends-by mon 12pm 17
domain, t4 ends-by tue 3pm 20
task, t1 1
task, t2 4
task, t3 1
task, t4 4
task, t5 3
task, t6 2
task, t7 3
task, t8 3
task, t9 1
task, t10 3
constraint, t6 same-day t5
constraint, t4 before t5
constraint, t10 after t1
constraint, t6 starts-at t1
domain, t5 ends-by wed 1pm 16
domain, t6 ends-by thu 3pm 20
domain, t3 ends-by thu 2pm 8
domain, t8 ends-by tue 10am 15
domain, t9 ends-by wed 2pm 7
task, t1 2
task, t2 3
task, t3 2
task, t4 3
task, t5 1
task, t6 4
task, t7 2
task, t8 3
task, t9 4
task, t10 1
constraint, t2 before t4
constraint, t6 same-day t1
constraint, t8 same-day t3
constraint, t1 before t2
domain, t8 ends-before fri 10am
domain, t7 ends-by thu 12pm 10
domain, t3 ends-by mon 1pm 17
domain, t5 ends-by fri 12pm 15
```

```
domain, t2 ends-by thu 1pm 5
domain, t10 ends-by fri 9am 13
domain, t9 ends-by tue 4pm 20
domain, t8 ends-by mon 3pm 9
domain, t1 ends-by mon 12pm 19
task, t1 2
task, t2 4
task, t3 4
task, t4 4
task, t5 1
task, t6 1
task, t7 2
task, t8 2
task, t9 1
task, t10 4
constraint, t8 before t1
constraint, t4 after t9
constraint, t4 same-day t7
constraint, t5 starts-at t9
domain, t4 starts-before wed 12pm
domain, t6 ends-by fri 1pm 13
domain, t2 ends-by fri 11am 19
domain, t5 ends-by tue 4pm 8
domain, t1 ends-by mon 1pm 15
domain, t10 ends-by wed 11am 20
domain, t7 ends-by wed 9am 7
domain, t8 ends-by tue 10am 17
domain, t3 ends-by fri 10am 12
task, t1 4
task, t2 2
task, t3 1
task, t4 4
task, t5 3
task, t6 2
task, t7 3
task, t8 1
task, t9 2
task, t10 2
constraint, t8 after t4
constraint, t4 same-day t10
constraint, t4 starts-at t2
constraint, t8 same-day t1
constraint, t1 before t9
constraint, t2 after t8
domain, t10 ends-after thu 3pm
domain, t5 starts-before mon 1pm
domain, t3 ends-by thu 12pm 6
domain, t10 ends-by thu 11am 17
domain, t6 ends-by mon 9am 13
domain, t4 ends-by tue 11am 5
domain, t1 ends-by mon 9am 17
task, t1 4
task, t2 2
task, t3 1
task, t4 1
task, t5 3
task, t6 1
task, t7 4
task, t8 3
task, t9 4
```

```
task, t10 2
constraint, t8 starts-at t4
constraint, t9 same-day t1
constraint, t3 before t8
constraint, t4 before t5
constraint, t10 after t9
domain, t3 starts-after fri 9am
domain, t6 starts-after thu 9am
domain, t1 ends-by thu 10am 10
domain, t8 ends-by wed 12pm 7
domain, t4 ends-by wed 10am 15
domain, t3 ends-by wed 2pm 17
domain, t6 ends-by mon 1pm 9
domain, t10 ends-by fri 1pm 17
task, t1 2
task, t2 4
task, t3 2
task, t4 2
task, t5 3
task, t6 2
task, t7 4
task, t8 1
task, t9 1
task, t10 1
constraint, t3 after t8
constraint, t6 same-day t3
constraint, t9 same-day t2
constraint, t3 after t6
constraint, t9 same-day t3
domain, t5 ends-after mon 12pm
domain, t8 ends-by mon 2pm 5
domain, t1 ends-by fri 1pm 11
domain, t4 ends-by thu 11am 14
domain, t6 ends-by thu 10am 6
domain, t5 ends-by thu 11am 15
task, t1 1
task, t2 2
task, t3 3
task, t4 4
task, t5 4
task, t6 1
task, t7 2
task, t8 3
task, t9 4
task, t10 1
constraint, t9 after t8
constraint, t5 before t3
constraint, t4 starts-at t5
constraint, t4 starts-at t7
constraint, t7 after t2
constraint, t10 starts-at t3
domain, t3 starts-after tue 4pm
domain, t3 ends-by wed 2pm 7
domain, t6 ends-by fri 10am 18
domain, t4 ends-by tue 12pm 10
domain, t8 ends-by mon 9am 20
domain, t2 ends-by wed 4pm 18
domain, t7 ends-by mon 2pm 7
task, t1 1
task, t2 1
```

```
task, t3 1
task, t4 2
task, t5 3
task, t6 3
task, t7 4
task, t8 4
task, t9 4
task, t10 3
constraint, t6 same-day t7
constraint, t8 starts-at t10
constraint, t8 before t7
constraint, t8 before t3
constraint, t4 before t2
domain, t8 starts-after thu 3pm
domain, t2 ends-by fri 9am 8
domain, t1 ends-by wed 2pm 10
domain, t3 ends-by tue 12pm 17
domain, t7 ends-by wed 11am 5
domain, t4 ends-by thu 3pm 16
domain, t6 ends-by fri 9am 20
domain, t9 ends-by tue 4pm 13
domain, t5 ends-by tue 10am 6
task, t1 4
task, t2 2
task, t3 4
task, t4 3
task, t5 3
task, t6 2
task, t7 1
task, t8 4
task, t9 3
task, t10 4
constraint, t6 before t8
constraint, t10 starts-at t6
constraint, t6 after t9
domain, t2 ends-by tue 3pm 12
domain, t5 ends-by fri 3pm 10
domain, t6 ends-by tue 10am 5
domain, t1 ends-by tue 4pm 11
domain, t8 ends-by mon 12pm 8
domain, t10 ends-by fri 1pm 17
task, t1 4
task, t2 4
task, t3 3
task, t4 2
task, t5 2
task, t6 1
task, t7 4
task, t8 1
task, t9 1
task, t10 2
constraint, t6 same-day t9
constraint, t3 same-day t8
constraint, t2 before t5
domain, t3 starts-after fri 4pm
domain, t2 starts-before thu 3pm
domain, t9 ends-by wed 9am 19
domain, t6 ends-by tue 9am 19
domain, t8 ends-by thu 12pm 6
domain, t3 ends-by mon 2pm 12
```

```
domain, t2 ends-by fri 11am 20
domain, t10 ends-by thu 2pm 5
domain, t5 ends-by mon 12pm 12
domain, t4 ends-by fri 10am 6
task, t1 3
task, t2 2
task, t3 3
task, t4 2
task, t5 1
task, t6 2
task, t7 1
task, t8 1
task, t9 2
task, t10 2
constraint, t3 after t4
constraint, t6 after t4
constraint, t8 same-day t3
domain, t8 starts-after mon 1pm
domain, t9 ends-by wed 2pm 10
domain, t7 ends-by fri 4pm 17
domain, t2 ends-by tue 12pm 15
domain, t5 ends-by thu 9am 6
domain, t4 ends-by thu 9am 8
domain, t1 ends-by tue 4pm 10
domain, t6 ends-by wed 3pm 20
domain, t8 ends-by wed 2pm 16
task, t1 2
task, t2 1
task, t3 4
task, t4 2
task, t5 4
task, t6 2
task, t7 3
task, t8 3
task, t9 3
task, t10 2
constraint, t1 after t3
constraint, t5 before t7
constraint, t1 same-day t3
constraint, t8 same-day t2
constraint, t4 before t2
constraint, t2 before t5
domain, t6 ends-by mon 3pm 16
domain, t2 ends-by thu 3pm 16
domain, t10 ends-by tue 3pm 18
domain, t3 ends-by wed 12pm 5
domain, t9 ends-by fri 4pm 6
task, t1 1
task, t2 3
task, t3 4
task, t4 1
task, t5 1
task, t6 2
task, t7 3
task, t8 4
task, t9 1
task, t10 2
constraint, t6 same-day t8
constraint, t2 before t6
constraint, t1 before t3
```

```
constraint, t2 same-day t9
constraint, t2 starts-at t3
constraint, t2 after t1
domain, t9 starts-after fri 1pm
domain, t10 starts-after fri 9am
domain, t8 ends-by wed 10am 19
domain, t4 ends-by fri 11am 10
domain, t5 ends-by tue 11am 17
domain, t6 ends-by wed 12pm 14
domain, t7 ends-by mon 2pm 10
domain, t9 ends-by mon 10am 19
task, t1 1
task, t2 2
task, t3 3
task, t4 2
task, t5 3
task, t6 1
task, t7 4
task, t8 3
task, t9 1
task, t10 3
constraint, t1 before t7
constraint, t6 after t2
constraint, t9 after t2
constraint, t2 same-day t1
constraint, t3 after t7
constraint, t4 same-day t9
domain, t1 ends-before thu 10am
domain, t5 ends-by mon 12pm 10
domain, t6 ends-by thu 2pm 7
domain, t10 ends-by tue 1pm 15
domain, t3 ends-by tue 10am 18
domain, t2 ends-by wed 1pm 8
```