

11/2 DFN 작업 현황 - 알고리즘

대규모 로직 업데이트가 필요한 부분이라 수정이 조금 오래 걸렸습니다. 코드가 갈수록 복잡해지고 있는데요. 일단 기능 구현에 집중한 뒤에 추후 가독성을 보강하겠습니다.

1. divide_nurse_info_by_team

`get_nurse_info` 와 `get_last_schedule` 함수의 출력값을 받아 스케줄을 짜는 데에 필요한 정보 형태로 가공한다.

1) 매개변수

(1) team_list

스케줄 제작에 필요한 팀의 번호들. `list` 형태로 입력.

(2) nurse_profile_dict

view함수의 `get_nurse_info` 의 출력값.

(3) nurse_last_months_schedule_dict

view 함수 중 `get_last_schedule` 의 출력값.

2) 반환값

딕셔너리.

(1) key

팀의 번호들

(2) value:

딕셔너리. 각 팀의 간호사 세부 정보를 담고 있다.

간호사pk: [간호사 일련번호, 간호사 등급, 팀 번호.... 정보가 기록됨.]

```
Key: 팀 번호들
value: 팀에 속하는 간호사들의 info 딕셔너리
0 NURSE_NUMBER 간호사 일련번호
1 NURSE_GRADE 간호사 grade
2 TEAM_NUMBER 팀넘버
3 SHIFTS, 이번 달 근무 일수
4 SHIFT_STREAKS, 연속 근무일 수
5 OFFS, 그.. 마크다운에 있는 'OFF' 참조.
6 MONTHLY_NIGHT_SHIFTS, 한 달에 night 근무한 횟수
7 VACATION_INFO, 휴가 정보(외부 딕셔너리로 수정 예정)
8 OFF_STREAKS, 연속 휴무
9 LAST_SHIFT, 마지막 근무 정보
```

```

def divide_nurse_info_by_team(
    team_list,
    nurse_profile_dict,
    nurse_last_months_schedule_dict
):
    # 1. 선언
    # 출력을 위한 dict 선언.
    divided_nurse_info_dict_by_team = dict()
    for team_number in team_list:
        divided_nurse_info_dict_by_team[team_number] = dict()

    # 2. 연산
    for nurse_detail in nurse_profile_dict:
        # 1) 앞부분에 필요한 값들.
        nurse_pk = nurse_detail[0]
        nurse_grade = nurse_detail[1]
        nurse_team = nurse_detail[2]
        nurse_offs = nurse_detail[3]

        # 2) 뒷부분에 필요한 값들.
        nurse_last_month_schedule = nurse_last_months_schedule_dict[nurse_pk]
        nurse_last_shift = nurse_last_month_schedule[-1]
        nurse_off_streaks = 0

        # 3) 딕셔너리에 삽입
        divided_nurse_info_dict_by_team[nurse_team][nurse_pk] = [
            nurse_pk,
            nurse_grade,
            nurse_team,
            0,
            nurse_offs,
            0,
            0,
            nurse_off_streaks,
            nurse_last_shift
        ]

    return divided_nurse_info_dict_by_team

```

2. make_monthly_schedule 수정사항

1) '팀별' 스케줄 제작 및 병동별 유효성 검사 체크.

(1) 스케줄 제작에 필요한 정보를 팀별로 파싱.

`divide_nurse_info` 함수 제작. 팀별로 나눠서 스케줄을 짤 수 있게 되었다.

이전:

```
ideal_schedule = make_ideal_counter(needed_nurses_shift_by_team)
```

이후:

```

ideal_schedule = make_ideal_counter(needed_nurses_shift_by_team)
divided_nurse_info = divide_nurse_info_by_team(
    team_list=team_list,
    nurse_profile_dict=dict(),
    nurse_last_months_schedule_dict=dict()
)

```

(2) 전반적 로직 변경.

shift 당 최소한 한 명 이상의 종연차 이상 간호사가 필요하다. 해당 로직을 구현하기 위해 함수가 수정되었다.

수정 이전

리스트 형태의 info 를 받아 팀 단위로 스케줄을 작성한다.

```

while current_date != MONTHS_LAST_DAY[current_month] + 1:

    # 2. 선언
    whole_team_temp_schedule = []

    while not is_enough_grade and recursion_time < 15:
        temporary_schedule, grade_counter_bit = make_daily_schedule(
            nurse_pk_list = nurse_pk_list,
            nurse_info = nurse_info,
            ideal_schedule = ideal_schedule,
            current_date = current_date
        )

```

수정 이후

1. 각 팀의 정보를 토대로 팀별 시간표를 제작한다.
2. `make_daily_schedule` 은 팀별 시간표와 검증 토큰을 함께 반환한다.
3. 비트마스킹 형태로 각 근무(day, evening, night) 당 grade가 1 이상인 간호사가 있는지 확인한다.

```

# 2. 선언
# 비트마스킹 형태로 grade의 참여 여부 확인.
whole_team_temp_schedule = []
is_enough_grade = False

while not is_enough_grade and recursion_time < 15:

    teamed_up_schedule = dict()
    whole_team_grade_checker = 1

    for team_number in team_list:
        temporary_schedule, grade_counter_bit = make_daily_schedule(
            nurse_pk_list = nurse_pk_list,
            nurse_info = divided_nurse_info[team_number],
            ideal_schedule = ideal_schedule,
            current_date = current_date
        )

        teamed_up_schedule[team_number] = temporary_schedule
        whole_team_grade_checker |= grade_counter_bit

```

```
if whole_team_grade_checker > 1:
    whole_team_temp_schedule.append(teamed_up_schedule)
    is_enough_grade = True

else:
    recursion_time += 1
```

3.기타 수정 예정

스케줄 구현의 전반적인 로직 수정.

nurse_info에 사용하는 변수들 수정 및 교체.